Université Paris Cité

LIPADE
Laboratoire d'Informatique PAris DEscartes

Université Paris Cité

École Doctorale EDITE (ED 130)
Laboratoire LIPADE (EA 2517)

Representation Learning for Multivariate Time Series

Thèse de doctorat en Intelligence Artificielle

Par Romain ILBERT

Dirigée par **Themis PALPANAS**

Présentée et soutenue publiquement le 16/05/2025

Composition du Jury:

Patrick Gallinari Distinguished Researcher, Criteo AI Lab & Professor, Sorbonne University	Président
Laurent Oudre Full Professor, ENS Paris-Saclay	Rapporteur
Elisa Fromont Full Professor, University of Rennes	Rapportrice
Marianne Clausel Full Professor, University of Lorraine	Examinatrice
Hubert Banville Research Scientist, Meta	Examinateur
Themis Palpanas Distinguished Professor, Paris-Cité University & LIPADE Director	Directeur
Ievgen Redko Principal Research Scientist, Huawei Noah's Ark Lab	Invité

Title : Representation Learning for Multivariate Time Series

Keywords: multivariate time series, representation learning, time series forecasting, time series classification, optimization, regularization, multi-task learning, transformers, foundation models, fine-tuning

Abstract:

This thesis introduces a unified framework for multivariate time series analysis by intertwining three complementary contributions. First, we present SAMformer—a novel, shallow transformer architecture tailored for time series forecasting. Unlike in natural language processing or computer vision, standard transformer models often underperform in forecasting tasks due to their tendency to converge to sharp minima during training, which impairs generalization. By integrating sharpness-aware minimization with a new channel-wise attention mechanism, SAMformer overcomes these pitfalls and achieves state-of-the-art performance. Second, in the context of multivariate forecasting, we observe that most existing methods predict each channel independently, thereby neglecting valuable inter-channel correlations. To address this, we develop a multi-task learning approach that incorporates a specialized regularization term into the training loss. This term is applied to a final layer that fuses independent channel outputs into a coherent multivariate prediction, effectively exploiting the shared dynamics among channels. Third, we tackle limitations in foundation models, which typically process time series in a univariate manner for tasks such as multivariate classification. We propose the design of intelligent adapters that combine information across channels, dramatically reducing computational time and memory usage while preserving predictive accuracy. Collectively, these contributions provide a coherent strategy for learning robust representations of multivariate time series.

Title : Apprentissage de représentations pour les séries temporelles multivariées.

Mots-clés : séries temporelles multivariées, apprentissage des représentations, prévision de séries temporelles, classification de séries temporelles, optimisation, régularisation, apprentissage multitâche, transformers, modèles fondamentaux, affinage

Résumé :

Cette thèse présente un cadre unifié pour l'analyse des séries temporelles multivariées en articulant trois contributions complémentaires. Tout d'abord, nous introduisons SAMformer — une nouvelle architecture de transformer peu profonde spécialement conçue pour la prévision des séries temporelles. Contrairement aux applications en traitement du langage naturel ou en vision par ordinateur, les modèles transformer classiques obtiennent souvent des performances insuffisantes en prévision, en raison de leur tendance à converger vers des minima aigus lors de l'entraînement, ce qui nuit à leur capacité de généralisation. En intégrant une stratégie d'optimisation sensible à la netteté (sharpness-aware minimization) avec un nouveau mécanisme d'attention par canal, SAMformer contourne ces écueils et atteint des performances de pointe. Ensuite, dans le contexte de la prévision multivariée, nous avons constaté que la plupart des méthodes existantes effectuent des prédictions univariées pour chaque canal, négligeant ainsi les corrélations spatio-temporelles essentielles entre les canaux. Pour y remédier, nous développons une approche d'apprentissage multitâche qui intègre un terme de régularisation spécifique dans la fonction de coût d'entraînement. Ce terme est appliqué à une couche finale qui fusionne les sorties issues de canaux indépendants en une prédiction multivariée cohérente, exploitant efficacement les dynamiques communes. Enfin, nous abordons les limitations des modèles de fondation, lesquels traitent généralement les séries temporelles de manière univariée pour des tâches telles que la classification. Nous proposons la conception d'adaptateurs intelligents permettant de combiner de façon optimale l'information provenant de différents canaux, réduisant drastiquement le temps de calcul et les besoins en mémoire tout en préservant la précision prédictive dans des tâches de classification multivariée. Collectivement, ces contributions offrent une stratégie cohérente pour apprendre des représentations robustes des séries temporelles multivariées.

Résumé général de la thèse

Contexte et Motivations

Les séries temporelles multivariées jouent un rôle crucial dans de nombreux domaines tels que la maintenance prédictive, la santé, la finance, l'énergie ou encore l'analyse climatique. Dans le contexte industriel, ces données permettent notamment la détection précoce de pannes, réduisant ainsi les coûts opérationnels et améliorant la sécurité des systèmes. En santé, elles sont essentielles au suivi en continu de l'état des patients, en particulier dans les unités de soins intensifs où une intervention rapide peut sauver des vies. Dans le secteur financier, l'analyse précise des séries temporelles est indispensable pour la gestion des risques et la prise de décisions d'investissement stratégiques. Enfin, pour l'analyse climatique, elles sont fondamentales à l'étude des tendances à long terme, contribuant à la compréhension et à la prédiction du changement climatique.

Toutefois, l'exploitation efficace de ces séries temporelles se heurte à plusieurs difficultés importantes. Premièrement, ces données présentent souvent des interactions complexes, ponctuelles et dynamiques entre leurs différentes variables, rendant leur analyse particulièrement difficile. De plus, elles sont fréquemment affectées par du bruit important, une forte non-stationnarité liée à des changements dans les processus sous-jacents, ainsi que par des données manquantes dues à des capteurs défectueux ou des interruptions de transmission.

Un autre défi majeur concerne la rareté des données labellisées disponibles pour entraîner des modèles prédictifs efficaces. Cette situation est particulièrement fréquente dans des domaines sensibles comme la médecine, où les questions de confidentialité limitent l'accès à de larges ensembles de données annotées. Par ailleurs, dans des contextes critiques tels que la santé ou la finance, il est impératif de disposer de prédictions en temps réel avec une très faible latence, tout en assurant une grande transparence et interprétabilité des modèles utilisés afin de garantir la confiance des utilisateurs finaux et la conformité aux régulations en vigueur.

Ainsi, bien que les récentes avancées en apprentissage automatique, et particulièrement dans les réseaux neuronaux profonds comme les transformers, aient révolutionné des domaines tels que le traitement du langage naturel et la vision par ordinateur, leur succès n'est pas encore pleinement transposé aux séries temporelles multivariées. Cela souligne un besoin critique de développer des modèles spécialisés capables de prendre en compte explicitement la structure complexe des interactions entre variables tout en restant robustes aux défis inhérents à ce type de données.

Lacunes des Approches Actuelles

Bien que les dernières années aient vu émerger des avancées notables dans la modélisation des séries temporelles, plusieurs limites importantes subsistent dans les approches actuelles :

Haute Dimensionnalité. Les séries temporelles multivariées modernes impliquent souvent un très grand nombre de variables, notamment dans l'industrie avec l'Internet des ob-

jets (IoT), la finance ou encore la surveillance médicale, où des dizaines voire des centaines de capteurs ou d'indicateurs doivent être traités simultanément. Les approches traditionnelles telles que les variantes d'ARIMA ou les réseaux neuronaux récurrents simples sont généralement incapables de gérer efficacement cette dimensionnalité élevée, entraînant soit une simplification excessive du modèle, soit un surapprentissage rapide. Même les modèles avancés comme les transformers peinent parfois à surpasser des modèles linéaires simples dans ces conditions, soulignant la nécessité de méthodes spécifiquement adaptées à la haute dimensionnalité.

Passage à l'échelle. Bien que les modèles de fondation aient démontré des performances remarquables en NLP et en vision par ordinateur, leur application directe aux séries temporelles multivariées se heurte souvent à des contraintes computationnelles sévères. Ces modèles nécessitent typiquement d'importantes ressources en mémoire et en temps de calcul, les rendant difficilement exploitables en temps réel sur des systèmes aux ressources limitées. Par ailleurs, ces modèles sont souvent conçus pour fonctionner sur des données massives, alors que les séries temporelles réelles disponibles dans les domaines sensibles sont fréquemment limitées en taille, nécessitant ainsi des approches de réduction dimensionnelle efficaces et adaptées.

Fondements Théoriques et Optimisation. Malgré leur puissance expressive, les transformers et autres architectures complexes rencontrent des difficultés à généraliser efficacement aux séries temporelles réelles en raison de problèmes liés aux paysages d'optimisation complexes et à la difficulté d'atteindre des minima généralisables. Ce phénomène est d'autant plus marqué dans des scénarios où les données sont limitées ou bruitées. De plus, l'absence d'un cadre théorique rigoureux permettant de comprendre clairement l'impact des caractéristiques des données (dimensionnalité, bruit, non-stationnarité) sur la performance des modèles limite l'amélioration systématique des méthodologies existantes.

Architectures Efficaces. Les architectures actuelles pour les séries temporelles souffrent souvent d'une complexité excessive qui empêche leur déploiement en temps réel dans des contextes sensibles où les décisions doivent être immédiates, comme en finance ou en santé. Cette complexité implique souvent des compromis difficiles entre précision prédictive, efficacité computationnelle et interprétabilité. La conception d'architectures légères, modulaires et adaptables, capables de préserver l'essentiel de la performance tout en étant suffisamment rapides et interprétables, demeure donc un enjeu majeur non résolu par les approches existantes.

Ces lacunes motivent pleinement les contributions de cette thèse, qui vise à apporter des solutions concrètes, robustes et innovantes, combinant une compréhension théorique approfondie avec des approches pratiques efficaces pour mieux répondre aux besoins réels de l'analyse des séries temporelles multivariées.

Contributions Principales

Face à ces défis, cette thèse apporte trois contributions majeures :

SAMformer : une variante robuste du Transformer. Dans le chapitre 3, nous explorons pourquoi les transformers, malgré leur puissance expressive démontrée en NLP et en vision par ordinateur, échouent souvent à surpasser des modèles linéaires plus simples en prévision multivariée à long terme. À partir d'un problème linéaire élémentaire, nous montrons que les transformers souffrent d'une mauvaise généralisation liée à la structure de leurs mécanismes d'attention, qui les conduit à des minima locaux trop pointus. Nous introduisons alors SAMformer, un Transformer léger utilisant la minimisation consciente de la netteté (SAM) et une attention spécifique par canal. Cette combinaison permet au modèle de mieux gérer l'instabilité lors de l'entraînement, favorisant la convergence vers des minima plats avec une généralisation améliorée. SAMformer intègre également des pratiques récentes telles que la normalisation réversible d'instance (RevIN), optimisant ainsi les performances en prévision à long terme sur plusieurs jeux de données réels couramment utilisés. Nos résultats expérimentaux montrent que SAMformer atteint l'état de l'art, ainsi que des performances équivalentes au modèle de fondation MOIRAI tout en ayant considérablement moins de paramètres.

Cadre de régularisation multi-tâches pour séries multivariées. Le chapitre 4 aborde la prévision des séries temporelles multivariées sous l'angle de l'apprentissage multi-tâches. Nous considérons chaque canal d'une série multivariée comme une tâche distincte, permettant ainsi une meilleure exploitation des informations partagées entre canaux. Nous proposons une stratégie d'optimisation innovante qui introduit une régularisation explicite encourageant l'apprentissage conjoint entre tâches tout en préservant des spécificités individuelles importantes. Nous fournissons également un cadre analytique détaillé permettant de comprendre comment équilibrer efficacement les composantes communes et spécifiques des séries temporelles. Ce cadre théorique est accompagné d'une méthode pratique fondée sur les statistiques des données, facilitant ainsi la sélection optimale des hyperparamètres. Nos évaluations empiriques montrent une nette amélioration des performances prédictives par rapport aux approches mono-tâches traditionnelles et une compétitivité marquée vis-à-vis des modèles multivariés avancés tels que SAMformer ou iTransformer.

Adaptation efficace des modèles de fondation aux séries temporelles. Le chapitre 5 traite du défi de rendre accessibles les modèles de fondation très performants mais gourmands en ressources dans le contexte des séries temporelles multivariées. Nous proposons une stratégie originale de compression de l'espace latent, réduisant ainsi drastiquement les exigences en mémoire et en temps de calcul tout en maintenant un niveau élevé de précision de classification. À travers des expériences approfondies, nous démontrons que notre approche permet de réduire l'espace latent à seulement 2,10% de sa taille initiale tout en préservant 96,15% des performances originales du modèle complet. De plus, nous explorons divers adaptateurs basés sur des méthodes classiques et des réseaux neuronaux afin d'optimiser davantage cette représentation compressée. Nos résultats soulignent un gain de vitesse jusqu'à dix fois supérieur par rapport aux modèles de référence, et permettent d'accueillir un nombre beaucoup plus important de jeux de données sur un seul GPU standard, rendant ainsi ces modèles de fondation pratiques pour un usage généralisé.

Ces contributions constituent un avancement majeur en modélisation des séries temporelles, répondant efficacement aux défis pratiques et théoriques du domaine.

Aperçu de la Thèse

Nous détaillons ci-après le contenu de cette thèse:

Le chapitre 1 constitue une introduction générale présentant le problème traité, exposant le contexte scientifique et industriel, et identifiant les motivations, les défis et les lacunes existantes dans les approches actuelles. Il précise également les objectifs de recherche et donne un aperçu des contributions majeures apportées par cette thèse.

Le chapitre 2 présente une revue de l'état de l'art relatif aux séries temporelles multivariées, en se concentrant sur trois domaines fondamentaux : la classification, la prévision et les modèles de fondation. Ce chapitre établit également le lien entre ces domaines et les objectifs spécifiques de la thèse, et pose les bases nécessaires à la compréhension des contributions proposées dans les chapitres suivants.

Le chapitre 3 présente en détail SAMformer, une variante légère et efficace des transformers, spécifiquement conçue pour pallier les limites constatées dans les séries temporelles. Nous y analysons les raisons pour lesquelles les transformers classiques échouent souvent en prévision à long terme, puis détaillons comment notre modèle intègre la minimisation consciente de la netteté (SAM) avec une attention spécifique par canal. Ce chapitre inclut une exploration approfondie de l'impact de ces choix méthodologiques sur la stabilité de l'entraînement et la performance en généralisation, validée par des expériences rigoureuses sur de nombreux jeux de données réels.

Le chapitre 4 introduit et développe un cadre innovant de régularisation multi-tâches appliqué à la prévision des séries temporelles multivariées. Nous formulons théoriquement le problème de prévision multivariée comme une collection de tâches interconnectées, en mettant en avant une régularisation capable d'exploiter les similitudes inter-canaux tout en respectant leurs particularités individuelles. Ce chapitre détaille les aspects théoriques et pratiques, notamment la sélection guidée par les statistiques des données pour les hyperparamètres, et démontre par des résultats expérimentaux solides l'efficacité accrue de cette approche comparée aux méthodes traditionnelles.

Le chapitre 5 traite du défi majeur de rendre les modèles de fondation accessibles et pratiques pour l'analyse de séries temporelles multivariées, grâce à une stratégie avancée de compression de l'espace latent. Nous étudions diverses techniques de réduction dimensionnelle permettant de préserver une grande partie de la performance originale tout en diminuant significativement les coûts computationnels et mémoires. Le chapitre explore en détail les performances obtenues en classification multivariée avec différentes méthodes d'adaptation, montrant un gain substantiel en rapidité d'entraînement et en capacité à gérer efficacement davantage de données sur un même matériel.

Le chapitre 6 conclut cette thèse en résumant les contributions clés et en proposant une réflexion sur les perspectives ouvertes par ce travail. Il présente des pistes pour les recherches futures, telles que l'extension anisotropique de SAM ou une factorisation matricielle pour de la prévision multi-échelle. Ce chapitre souligne l'importance de poursuivre des travaux interdisciplinaires pour répondre aux défis complexes posés par les séries temporelles multivariées.

«J'avais appris que la patience était une vertu suprême, la plus élégante et la plus oubliée. Elle aidait à aimer le monde avant de prétendre le transformer. Elle invitait à s'asseoir devant la scène, à jouir du spectacle, fût-il un frémissement de feuille. La patience était la révérence de l'homme à ce qui était donné.»

Sylvain Tesson, *La panthère des neiges*.

Acknowledgements

Trois années se sont écoulées, trois années qui sont passées particulièrement vite, notamment dans la dernière partie de la thèse. Ce parcours a été un peu atypique pour moi, car j'ai eu l'occasion de travailler dans deux équipes très différentes chez Huawei. La première équipe était orientée business, ce qui rendait difficile la publication de résultats. Par la suite, j'ai eu la chance de rejoindre une équipe davantage tournée vers la recherche, où j'ai bénéficié d'une très grande liberté sur les sujets que je voulais explorer. C'est dans cette deuxième équipe que je me suis réellement épanoui, entouré de chercheurs toujours disponibles, ouverts à l'échange et aux conseils précieux.

Je souhaite remercier chaleureusement Themis Palpanas, mon directeur de thèse, qui m'a accompagné tout au long de ce parcours, et sans qui rien de tout cela n'aurait été possible. Merci Themis pour tes précieux conseils, ton accompagnement constant, et la grande liberté que tu m'as accordée dans le choix des sujets qui me passionnaient. Ces années de recherche ont été épanouissantes pour moi, en grande partie grâce à toi. J'ajoute aussi que nos échanges, souvent teintés d'humour, ont apporté une agréable légèreté tout au long de ce travail.

Je tiens également à remercier Ievgen Redko, qui a co-encadré la seconde partie de ma thèse aux côtés de Themis. Merci Ievgen de m'avoir offert autonomie et confiance dans mes travaux de recherche, ainsi que pour tes conseils toujours pertinents.

Je remercie également mes encadrants de Huawei durant la première année et demie de ma thèse, Zonghua et Thai, ainsi que tous les membres de l'équipe Autonomous Driving Networks.

Enfin, je remercie chaleureusement l'ensemble des membres de l'équipe Noah's Ark pour nos échanges toujours constructifs, avec une mention particulière pour Malik, pour nos nombreuses discussions mathématiques, mais aussi nos échanges plus légers sur le foot, qui ont rendu ces moments particulièrement agréables et également pout Vasilii, avec qui j'ai beaucoup discuté autour des foundation models.

Je tiens à également exprimer toute ma gratitude à Alexandre Gramfort, pour avoir participé à mes deux comités de suivi, pour ses conseils précieux et nos échanges. Merci également à toute l'équipe DiNo du LIPADE pour ces trois années enrichissantes, et plus particulièrement à Adrien, Qitong, Sijie, Christos, Manos, Mohammed et Paul.

Je voudrais aussi remercier une personne en particulier, celle qui a été là depuis le tout début, dans les moments difficiles comme dans les moments de joie, celle qui m'a

vu évoluer, et qui a toujours été là pour moi avec bienveillance et intelligence : ma femme, Sophie. Son soutien moral indéfectible pendant ces années de travail intense et de bouleversements administratifs a été fondamental. Ce manuscrit n'existerait pas sans elle. Je suis le plus heureux qu'elle soit devenue ma femme pendant cette thèse, et je dédie également ce manuscrit à notre futur enfant. Je garderai toujours en mémoire les merveilleux voyages que nous avons faits pendant ces années, qui ont rendu cette période de ma vie idyllique, comme tout ce que je vis à tes côtés depuis que je te connais.

Je tiens également à remercier mes amis proches pour leur présence, leur écoute et leur soutien tout au long de cette thèse. Un merci tout particulier à Mathis, pour ses conseils toujours avisés et nos échanges, qu'ils soient scientifiques ou non. Je remercie aussi chaleureusement Margaux et Aaron pour nos discussions enrichissantes, ainsi que Lucas, Jules, Maxime, Foulques, Valentin, Victor, Nicolas, Sofiane avec qui j'ai partagé tant de moments précieux au fil de ces années.

Enfin, je souhaite remercier toutes les personnes avec qui j'ai eu le plaisir d'échanger au fil des meetings, séminaires, summer schools ou conférences. Je pense en particulier à mon ancien professeur Marco Cuturi, ainsi qu'à ses collègues Pierre et Eugène, à Samy Bengio, Lucas Beyer, Cédric Rommel, Vianney Perchet, mais aussi à Soojung, Hermina, Mehdi, Sam, Benjamin et Hugues, pour la richesse de nos échanges. Je remercie également Kashif Razul pour nos discussions sur X, ainsi que pour sa contribution active à l'intégration de SAMformer dans GluonTS.

Je remercie aussi chaleureusement les membres de mon jury : Laurent Oudre, Elisa Fromont, Marianne Clausel, Patrick Gallinari et Hubert Banville, pour avoir accepté d'évaluer mon travail. Je suis honoré d'avoir l'opportunité de discuter de mes recherches en leur présence. Je remercie tout particulièrement Laurent Oudre et Elisa Fromont pour leurs retours sur mon manuscrit.

Enfin, un merci sincère à Stéphane Mallat pour ses cours profondément inspirants donnés au Collège de France.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 What this thesis tries to address

1.1.1 Motivation

Time series data arise in a wide range of fields—including predictive maintenance, healthcare, finance, and climate modeling—where analysts aim to uncover actionable patterns. These patterns may involve detecting faults early in industrial systems, monitoring patients in a hospital, guiding decisions in financial markets, or analyzing climate trends over long time spans. Despite the natural fit of time series for tracking changes over time, these data often involve many variables, contain substantial noise, and exhibit intricate dependencies across different channels. Consequently, many machine learning algorithms struggle to effectively leverage the rich structure of *multivariate* time series, especially when only limited data are available.

Challenges in modeling multivariate time series are plentiful. First, capturing the *complex interactions* among multiple variables is inherently difficult. These interactions can be sporadic—some variables may correlate only in specific time windows and remain uncorrelated otherwise—and they can be linear or non-linear in nature. Physical constraints add yet another layer of complexity. For instance, one sensor might influence another with a delay, making it hard to detect the underlying dependencies. All these factors complicate our ability to model and interpret these interactions.

Second, many real-world time series are both *non-stationary and noisy*, meaning the underlying data-generating processes can shift over time, while sensor failures and missing data can create outliers and gaps. As an example, energy consumption data typically exhibit *non-stationary* patterns due to seasonal variations and user behavior, and these data may also contain measurement errors.

Another difficulty arises from the frequent *scarcity of labeled data*, whether because of privacy regulations (e.g., in the medical domain) or simply because critical events are rare.

Brain signals like EEG (electroencephalography) recordings, for instance, are often under-supervised due to the high cost and time required for expert annotation, as well as ethical issues surrounding data sharing. Such scarcity of labels can hamper the training and validation of accurate models. Meanwhile, high-stakes sectors such as finance or healthcare increasingly demand *interpretability*, to build experts' trust and meet regulatory requirements, as well as *low-latency* predictions to enable real-time decisions or interventions. In healthcare, for example, clinicians in intensive care units rely on immediate predictions of patient deterioration and must understand why a model raises an alert.

Although powerful architectures such as transformers (Vaswani et al., 2017) dominate large-scale language and computer vision tasks, these approaches do not always perform as well in time series. In some cases, they can even be outperformed by simpler linear models. Moreover, these architectures often assume independence among variables, contradicting the fundamental idea behind multivariate time series, where leveraging inter-variable relationships is key. By neglecting these dependencies, such models cannot fully capture the richness of multivariate time series data.

Despite major advancements in machine learning, particularly in deep networks and sequence-based architectures, most current methods still do not sufficiently address the combination of high dimensionality, noise, non-stationarity, sparse labeling, interpretability, and low-latency constraints characteristic of real-world multivariate time series.

Against this backdrop, this thesis is driven by the need to develop *robust, efficient, scalable* methods and intuitive insights for modeling multivariate time series. We draw on recent advances in multitask learning to share relevant information across multiple tasks or channels, thereby boosting generalization. We also look to insights from transformer-based approaches—particularly regarding how they capture long-range dependencies—and adapt them for time series by carefully controlling complexity and mitigating overfitting. In addition, we explore how large-scale foundation models may be leveraged in time series, provided we introduce efficient adapters that keep computations tractable. Altogether, the goal of this thesis is to bridge the gap between theoretical insights and practical solutions, yielding approaches that handle real-world constraints—limited data, noisy signals and interpretability needs—while delivering state-of-the-art performance.

1.1.2 Research Gaps

Although significant progress has been made in time series modeling, several gaps persist in current methodologies, preventing their widespread adoption and reliable performance in real-world multivariate scenarios:

1. High Dimensionality. Modern time series applications often involve tracking dozens, if not hundreds, of variables in tandem, whether in industrial IoT (e.g., temperature, pressure, flow rates) or in healthcare and finance (e.g., multiple physiological signals, market indicators). Traditional models, such as ARIMA variants (G. E. Box & G. M. Jenkins, 1970)

or basic recurrent networks (Hochreiter & Schmidhuber, 1997a), become overwhelmed by this fast-rising dimensionality, leading to overfitting, excessive computation, or simplistic assumptions that treat each channel as independent. Meanwhile, even more advanced transformer-based architectures (H. Zhou et al., 2021; H. Wu et al., 2021) can sometimes be outperformed by naïve predictors in certain tasks (Guan Lai et al., 2018), underscoring persistent methodological gaps in time series forecasting. Combining multiple channels can enhance accuracy by leveraging interactions between channels. For instance, when pressure changes lag behind temperature changes, or when multiple financial indicators collectively hint at market shifts. Multi-task or multi-channel approaches (Caruana, 1997) can help unify these signals, yet they remain underexplored or heuristically applied. We assume that a more principled integration of inter-variable information, one that balances model complexity and interpretability, could substantially improve forecasting, classification, and more generally multivariate time series representations across diverse domains.

2. Scalability. Large-scale *foundation models* have transformed natural language processing and computer vision, and they are increasingly being applied to time series tasks (Ansari et al., 2024; Goswami et al., 2024; G. Woo et al., 2024a; Jin et al., 2024; T. Zhou, PeiSong Niu, et al., 2023; Das et al., 2024). However, directly using them for multivariate time series poses unique challenges. Real-world time series often offer limited labeled data, and their signals can be irregular, noisy, or prone to missing values, unlike the large, more curated datasets common in language and vision. Moreover, foundation models demand significant computational resources, creating bottlenecks in scenarios requiring near-real-time predictions, such as high-frequency financial trading or industrial anomaly detection. Similarly, many forecasting architectures have become disproportionately large, sometimes reaching hundreds of millions of parameters for relatively modest datasets (Gamboa, 2017; Oreshkin et al., 2020; H. Zhou et al., 2021; H. Wu et al., 2021; T. Zhou, Ma, et al., 2022; J. Nie et al., 2023; W. Woo et al., 2022; Y. Zhang et al., 2021). These models often employ heavy regularization to avoid overfitting, exposing the need for a more fundamental rethinking of model design to enhance data efficiency and resource usage (Gamboa, 2017; Oreshkin et al., 2020). A fundamental rethinking of scalability is therefore required, not just to accommodate increasing model sizes, but to develop more efficient approaches that enhance both performance and interpretability.

3. Theoretical Foundations and Optimization. While deep networks have seen success in other fields, time series often expose hidden limitations in their optimization and generalization capabilities. Transformers, for instance, are universal sequence approximators in theory (Yun et al., 2019), yet they frequently underperform simple linear baselines when confronted with real-world time series—raising the question of how to realize their theoretical promise (Guan Lai et al., 2018). Part of the challenge lies in suboptimal optimization landscapes, where regularization methods or specific weight initializations may fail to guide these complex architectures toward global minima that capture subtle spatio-temporal relationships (Bengio et al., 1994; Glorot & Bengio, 2010a) and generalize well. Another challenge is the lack of rigorous analytical frameworks that can shed light on

how factors like the number of samples, the dimensionality, or the noise levels may affect model behavior. Such theoretical insights are essential for developing more architectures that lead to greater robustness and interpretability.

4. Efficient Architectures. Although transformers have revolutionized sequence modeling through parallel self-attention, applying them directly to time series reveals multiple bottlenecks (H. Zhou et al., 2021). First, they can be data-hungry and prone to overfitting when faced with smaller or noisier datasets than those found in NLP or vision. Second, they often impose heavy computational and memory demands, which undermines real-time deployment in fields like finance, healthcare, or industrial control. Simply shrinking these architectures can strip away their ability to capture fine-grained spatio-temporal dependencies (Q. Wen et al., 2023). A more promising direction lies in designing selective, lightweight attention mechanisms, incorporating specific regularization methods, and building interpretable modules that highlight relevant features without incurring prohibitive overhead. In doing so, one can balance the expressive power of self-attention with the constraints of computational resources, data availability, and generalization.

In sum, these gaps underscore the need for substantial innovation in multivariate time series. There is a pressing requirement to better exploit cross-channel information, to improve the scalability and efficiency of existing architectures, and to overhaul current optimization practices so as to understand why powerful models like transformers—so successful in NLP and computer vision—can underperform linear baselines in multivariate forecasting tasks. A deeper understanding of these empirical and theoretical insights also enables more interpretable solutions, which prove essential in sensitive fields like healthcare, where trust and transparency are indispensable. Building on these insights, the subsequent chapters of this thesis introduce novel architectures and robust theory while leveraging multivariate information to learn better representations.

Connecting the Gaps to our Proposed Contributions. In order to address the shortcomings discussed in the preceding sections, this thesis puts forward three key contributions that combine theory and applicability.

Given the growing evidence that transformer-based architectures can sometimes underperform even naive baselines on real-world time series, we propose SAMformer. This lightweight yet robust transformer variant integrates *sharpness-aware minimization (SAM)* and channel-focused attention to alleviate overfitting and converge toward flat optima, which are known to enhance generalization. Unlike large-scale transformers that require massive datasets or resort to aggressive regularization, SAMformer emphasizes a stable training and optimization strategy to locate flat local minima that generalize effectively, even under domain shift. Empirical evaluations on multivariate forecasting tasks show that SAMformer achieves state-of-the-art performance while remaining the most lightweight model, making it an *efficient*, *robust* and *scalable* solution, especially when compared to foundation models such as MOIRAI (G. Woo et al., 2024a).

To better exploit cross-channel dependencies in high-dimensional time series, this the-

sis introduces a *multi-task regularization* framework. Many existing forecasting models assume that multivariate time series channels are independent, leading them to treat each channel separately. Our approach challenges this assumption by incorporating a regularized loss function that explicitly captures interdependencies across channels. We evaluate this framework on both linear and non-linear models, including transformer-based architectures. The regularization strength is controlled by a parameter λ, which we derive analytically. Our theoretical insights obtained from linear models extend well to non-linear architectures. The experiments reveal that even simple linear models trained with our regularized loss achieve performance comparable to state-of-the-art multivariate forecasting models on commonly used multivariate time series forecasting benchmarks. Our framework enhances *robustness*, improves *interpretability*, and effectively exploits multivariate dependencies.

While *foundation models* have demonstrated strong generalization capabilities in natural language processing and computer vision, their application to multivariate time series remains constrained by *computational inefficiency* and *scalability* challenges. This thesis introduces a novel approach to adapting foundation models efficiently for multivariate time series, leveraging *latent space compression techniques* to enhance computational feasibility while preserving performance. By structuring the adaptation process around resource constraints, we enable the deployment of powerful pre-trained models on real-world time series tasks, making them *more accessible* and *practical* for large-scale applications.

Taken together, these contributions provide a cohesive strategy for advancing time series analysis, balancing theoretical rigor, computational efficiency, and practical applicability in data- and resource-constrained environments.

1.2 Overview of the thesis

This thesis focuses on advancing the state of the art in multivariate time series analysis, with a particular emphasis on learning robust and scalable representations. By addressing critical challenges such as high dimensionality, complex dependencies, and data efficiency, it aims to contribute to both the theoretical understanding and practical application of time series modeling. Below, we provide an overview of the core objectives and contributions of this work.

Chapter 3: SAMformer: Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention

Despite the success of transformers in natural language processing and computer vision, transformer-based architectures often fall short in multivariate long-term forecasting, sometimes performing no better than simple linear models (Zeng, M. Chen, et al., 2023). In Chapter 3, we present SAMformer—a lightweight, robust transformer variant specifically designed to overcome these shortcomings. Our approach combines Sharpness-Aware

Minimization (SAM), a well-established method to drive optimization toward flat minima, with a novel channel-wise attention mechanism. This mechanism is designed to mitigate issues such as sharp loss landscape, which has been shown to impair the training stability of transformers (X. Chen et al., 2022; Zhai et al., 2023). By focusing on per-channel interactions rather than temporal attention, SAMformer reduces the parameter count and computational cost while significantly improving generalization. Empirical evaluations demonstrate that SAMformer not only converges more stably but also outperforms linear, mixer-based, transformer-based models and larger foundation models such as MOIRAI (G. Woo et al., 2024a), all while being markedly faster and more scalable.

Chapter 4: On Multi-Task Learning in Multivariate Time Series Forecasting

In high-dimensional forecasting problems, capturing the interdependencies among multiple time series channels is crucial. However, many existing approaches treat each channel in isolation, which can lead to overfitting and poor utilization of the shared structure inherent in the data. In Chapter 4, we propose a multi-task regularization framework that integrates an additional regularization term into the learning objective. This term is specifically designed to extract and exploit multivariate information by enforcing shared representations across channels. Our method can be easily integrated into existing neural network architectures. In practice, even linear models augmented with our regularization term are shown to outperform standard baselines—illustrating that our technique effectively leverages cross-channel information to enhance predictive performance.

Chapter 5: On Adapting Foundation Models to Multivariate Time Series Classification

While large foundation models have revolutionized fields like NLP and computer vision, their direct application to multivariate time series classification is often hindered by high computational demands and scalability issues. In Chapter 5, we propose a novel adapter framework that compresses the time-space representation through advanced latent space compression techniques. This adaptation makes it feasible to deploy large pre-trained foundation models in resource-constrained environments without sacrificing classification accuracy. Our approach is demonstrated using Mantis and MOMENT (Feofanov, S. Wen, et al., 2024; Goswami et al., 2024). The adapted model offers significant multivariate fine-tuning speedups while being on par with its full and non-adapted counterpart. This work thus bridges the gap between powerful pre-trained models and practical, large-scale applications in multivariate time series classification.

Together, these contributions address key limitations in existing approaches and lay the groundwork for future research in multivariate time series analysis. The following chapters provide a detailed exploration of each contribution, highlighting both the theoretical underpinnings and practical implementations.

1.3 Publications included in the thesis

This thesis is centered around key publications that have contributed to the field of multivariate time series analysis. These publications are presented in the main body of the thesis, as they represent the core contributions of this research:

Chapter 3

- Ilbert, R., Odonnat, A., Feofanov, V., Virmaux, A., Paolo, G., Palpanas, T., Redko, I. 2024. Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention. *International Conference on Machine Learning (ICML 2024) [Oral]*.

Chapter 4

- Ilbert, R., Feofanov, V., Tiomoko, M., Palpanas, T., Redko, I. 2024. Enhancing Multivariate Time Series Forecasting via Multi-Task Learning and Random Matrix Theory. *Time Series in the Age of Large Models (NeurIPS Workshop)*.
- Ilbert, R., Feofanov, V., Tiomoko, M., Odonnat, A., Palpanas, T., Redko, I. 2024. Analysing Multi-Task Regression via Random Matrix Theory with Application to Time Series Forecasting. *Advances in Neural Information Processing Systems (NeurIPS 2024) [SpotLight]*.

Chapter 5

- Ilbert, R., Feofanov, V., Tiomoko, M., Palpanas, T., Redko, I. 2025. User-friendly Foundation Model Adapters for Multivariate Time Series Classification. *Multivariate Time Series Analytics Workshop (International Conference on Data Engineering Workshop).*

1.4 Publications *not* included in the thesis

While the following publications are relevant to this research, they are not included in the main body of the thesis.

- Ilbert, R., V. Hoang, T., Zhang, Z., Palpanas, T. 2023. Breaking Boundaries: Balancing Performance and Robustness in Deep Wireless Traffic Forecasting. ARTMAN (ACM CCS Workshop).
- Ilbert, R., V. Hoang, T., Zhang, Z. 2024. Data Augmentation for Multivariate Time Series Classification: An Experimental Study. *Multivariate Time Series Analytics Workshop (International Conference on Data Engineering Workshop)*.

- Feofanov, V., Wen, S., Alonso, M., Ilbert, R., Guo, H., Tiomoko, M., Pan, L., Zhang, J., Redko, I. 2025. MANTIS: Foundation Model with Adapters for Multichannel Time Series Classification. *Technical Report*.

Chapter 2

Background and Related Work

Understanding how to effectively model multivariate time series is central to this thesis, particularly through the lens of *representation learning*, the task of discovering informative features from raw time series data. To set the stage for the contributions described in later chapters, we first review three key areas of existing literature: *time series classification*, *forecasting*, and *foundation models*. These areas are highly relevant as they represent common tasks in real-world applications, each providing different challenges and opportunities for representation learning. While classification focuses on assigning categorical labels to entire sequences, forecasting aims at predicting future values based on past observations, and foundation models explore generalizable large-scale representations. This chapter reviews foundational methods and recent advances in these three areas.

2.1 Definition and importance of time series

2.1.1 What is a time series?

A time series is a sequence of data points collected or recorded at successive time intervals, capturing how one or more variables evolve over time. Formally, we define a multivariate time series of length T as

$$\{\mathbf{x}_t\}_{t=1}^{T}, \quad \text{where } \mathbf{x}_t \in \mathbb{R}^D.$$

Each vector $\mathbf{x}_t$ consists of D channels (or features), with $\mathbf{x}_t = \left(x_{t,1}, \ldots, x_{t,D}\right)$. In matrix form, the entire series can be written as $\mathbf{X} \in \mathbb{R}^{T \times D}$, where the (t, d)-th entry, $\mathbf{X}_{t,d}$, corresponds to the value of the d-th channel at time t. An example of such a time series is depicted in Figure 2.1. Univariate time series naturally arise as the special case where $D = 1$.

Time series data are characterized by several key features:

- **Temporal dependency:** Observations are generally not independent; the value at

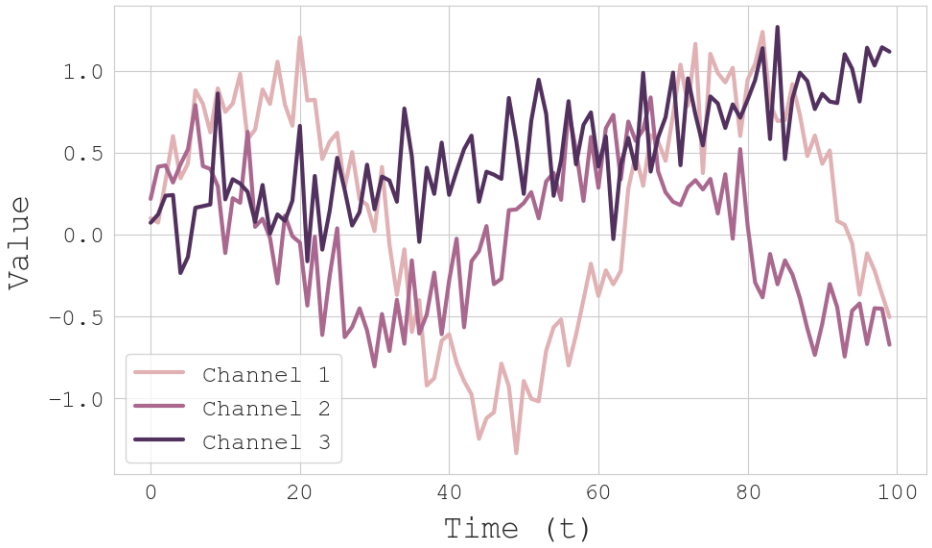


Figure 2.1: Example of a D-dimensional multivariate time-series with D=3.

a given time often depends on past observations.

- **Trend and seasonality:** Many time series exhibit long-term growth or decline patterns (trends), as well as periodic behaviors (seasonality).
- **Irregularities:** Time series data frequently contain noise, missing values, or abrupt shifts, making them more challenging to model than static datasets.
- **Channel interdependence (multivariate only):** In the multivariate setting ($D > 1$), different channels may have complex, time-varying, and potentially nonlinear relationships with one another. Identifying and leveraging these interdependencies is essential for improving performance but adds further complexity to the modeling design.

These characteristics make time series unique and necessitate specialized methods to capture their underlying temporal structure and cross-channel dependencies effectively.

2.1.2 Temporal Dependency

Temporal dependency refers to the intrinsic relationship between observations in a time series across different time steps. Unlike independent and identically distributed (i.i.d.) data, time series observations are sequentially ordered: the value at time t often depends on past values (lagged dependency) and can also influence future values (causal dependency).

Probabilistic Viewpoint. From a probabilistic perspective, a time series $\{x_t\}_{t=1}^{T}$ can be viewed as a realization of an underlying stochastic process. For i.i.d. data, the joint

probability distribution factorizes as

$$\mathbb{P}(x_1, x_2, \ldots, x_T) \;=\; \prod_{t=1}^{T} \mathbb{P}(x_t),$$

reflecting the absence of temporal or cross-sample dependencies. In contrast, time series data generally obey

$$\mathbb{P}(x_1, x_2, \ldots, x_T) \;=\; \prod_{t=1}^{T} \mathbb{P}\big(x_t \,\big|\, x_{t-1}, x_{t-2}, \ldots, x_1\big).$$

Hence, each observation x_t is potentially correlated with its past. In many practical scenarios, a simplified assumption known as the *Markov property* (or *finite-order Markov property*) is made, whereby

$$x_t \mid x_{t-1}, x_{t-2}, \ldots, x_1 \quad\approx\quad x_t \mid x_{t-1}, \ldots, x_{t-p},$$

meaning only the last p observations have a significant direct influence on the current value. This motivates the notion of a *model order* p, which encapsulates how far back in time the process $\{x_t\}$ looks when generating the next observation.

Parametric Model. Formally, one may view a time series $\{x_t\}_{t=1}^{T}$ as being generated by an underlying process

$$x_t = f_\theta\big(x_{t-1}, x_{t-2}, \ldots, x_{t-p}\big) + \varepsilon_t, \tag{2.1}$$

where f_θ is a potentially nonlinear function parameterized by θ, p denotes the order of temporal dependence, and ε_t is an error term, often assumed to be white noise with zero mean and finite variance. In a classical autoregressive (AR) model, f_θ takes a linear form,

$$\sum_{i=1}^{p} \phi_i \, x_{t-i},$$

where $\theta = (\phi_1, \phi_2, \ldots, \phi_p)$, with each $\phi_i \in \mathbb{R}$, corresponds to the set of AR coefficients (G. E. Box & G. M. Jenkins, 1970; Hamilton, 1994).

Short- vs. Long-Term Dependencies. Depending on the nature of the data, the relevant temporal dependency may manifest over short or long horizons (Brockwell & R. A. Davis, 2009). For instance, a one-step-ahead forecast in financial markets might rely on only a few recent observations, whereas physiological signals (e.g., electroencephalograms) may exhibit longer-term rhythms and periodicities (e.g., circadian cycles).

Look-Back Window. In modern forecasting practice—particularly in deep learning approaches—practitioners often prefer to specify the number of past time steps to consider, rather than a direct measure of temporal length. The parameter L, called the *look-back*

window, thus plays a role analogous to the order p discussed above, and is typically measured in discrete time steps. In state-of-the-art neural forecasting models, it is common to set $L = 336$ or $L = 512$ (H. Zhou et al., 2021; H. Wu et al., 2021), enabling the model to capture extended historical context. This approach is more flexible when the underlying data have varying sampling rates or when one wishes to experiment with different window lengths without redefining a notion of calendar time (e.g., daily vs. hourly). By tuning L, one can calibrate the trade-off between capturing long-range dependencies and managing computational complexity.

Challenges and Opportunities. A key challenge lies in identifying which past observations are relevant and how they influence future predictions. The dependency may be time-varying, nonlinear, and subject to regime shifts, trends, or seasonality. Moreover, in multivariate settings, cross-channel relationships add another layer of complexity (Tsay, 2013).

2.1.3 Trend and Seasonality

Trend and seasonality are structural components frequently observed in time series data.

- **Trend:** A trend represents the long-term increase or decrease in the values of a time series. It may result from external factors such as population growth, economic policies, or gradual technological advancements. Identifying trends helps to understand the overall direction of change over time.

- **Seasonality:** Seasonality refers to recurring patterns or cycles at regular intervals, such as daily, weekly, or yearly variations. For instance, energy consumption tends to follow seasonal patterns due to temperature changes, and retail sales often peak during holidays.

Both trend and seasonality can obscure the underlying dynamics of a time series, making their removal or modeling essential for accurate prediction or classification (Cleveland et al., 1990). Techniques such as decomposition are commonly employed to isolate these components. An example of such a decomposition is illustrated in Figure 2.2, where a time series is separated into its original signal, trend, and seasonal component.

Beyond these patterns, external factors—known as external covariates—can provide valuable additional information for forecasting. These covariates include variables that are known in advance, such as holidays, promotional events, or economic indicators, that can enhance predictive accuracy.

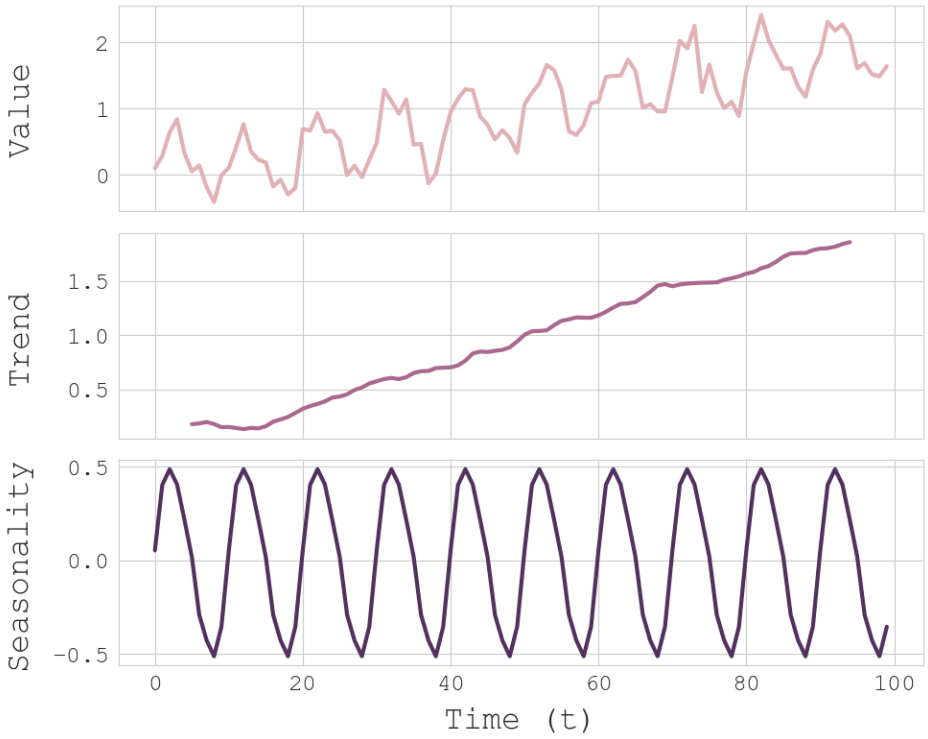


Figure 2.2: Breakdown of a channel into its fundamental components: original signal (top), underlying trend (middle), and seasonal variations (bottom)

2.1.4 Irregularities

Irregularities in time series encompass noise, missing values, and abrupt changes or anomalies that deviate significantly from the expected behavior (Little & Rubin, 2002; Chandola et al., 2009; Boniol et al., 2024). These can arise from various sources:

- **Noise:** Random fluctuations that do not contain meaningful information, often resulting from measurement errors or external disturbances.
- **Missing values:** Gaps in the recorded data due to hardware failures, transmission errors, or data corruption.
- **Anomalies:** Unexpected spikes, drops, or structural breaks, often indicative of critical events, such as equipment failures in industrial systems or financial crashes in stock markets. While anomalies can introduce noise into forecasting models, they can also serve as distinctive *signatures* of a time series, making them useful features for classification tasks. For instance, the occurrence of specific anomaly patterns in physiological signals can aid in the diagnosis of medical conditions.

Handling irregularities is essential for effective analysis. Noise can obscure meaningful patterns, while anomalies may lead to biased models if not accounted for properly. Robust preprocessing techniques, such as smoothing (Friedman, 1984), imputation (Little & Rubin,

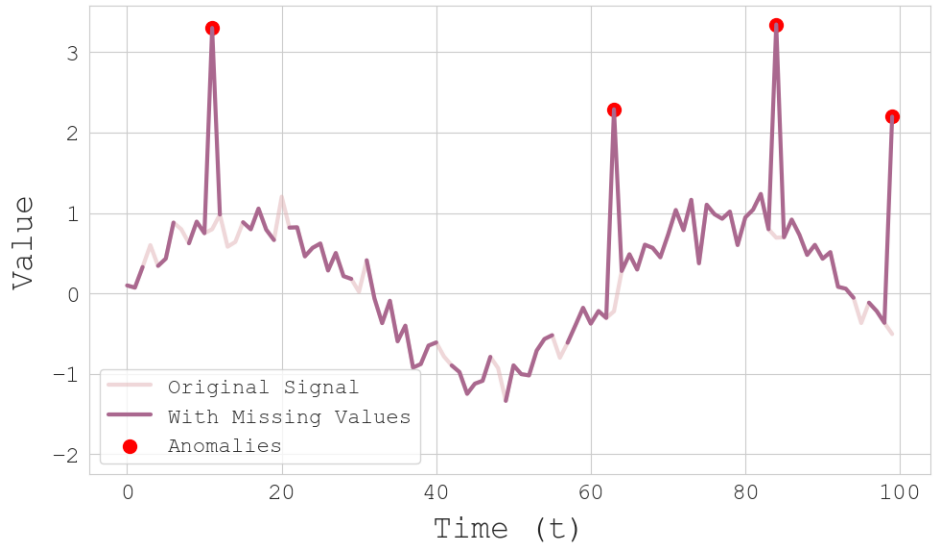


Figure 2.3: Illustration of irregularities. The plot highlights missing values (gaps) and anomalies (red circles), all of which can affect model performance if not properly addressed. However, anomalies may also serve as informative features for classification tasks.

2002), and outlier detection (Chandola et al., 2009), are commonly used to mitigate these effects. However, in some cases, anomalies carry valuable information that can be leveraged for classification problems. For instance, specific patterns of anomalies can serve as distinguishing features in shapelet-based classification methods (Ye & Keogh, 2009; Hills et al., 2014). Identifying recurrent irregularities across different time series can help distinguish between different categories or states, enhancing the robustness of classification models. An example of these irregularities—missing values and anomalies—can be seen in Figure 2.3.

2.1.5 Channel Interdependence in Multivariate Time Series

In multivariate time series, different channels (or features) often exhibit complex dependencies that evolve over time. Unlike univariate series, where each observation depends solely on its own past values, multivariate series involve interactions between multiple variables that can be linear or nonlinear, static or dynamic.

Defining Channel Interdependence. Unlike in the univariate case, where each value depends only on its own past, multivariate time series introduce interactions between different channels. This means that the value $x_{t,d}$ at time t is not only influenced by its own history but also by the past values of other channels. A simple way to represent this dependency is:

$$x_{t,d} = f_d(x_{t-1,1}, \ldots, x_{t-1,D}) + \varepsilon_t, \tag{2.2}$$

where f_d models the relationships between channels and ε_t represents noise. In practice, these dependencies are rarely static: they often evolve over time, meaning that f_d itself may change dynamically (Xuan & Murphy, 2007; Zhao & Shen, 2024). Moreover, in many real-world scenarios, dependencies go beyond just the previous time step. Instead of relying solely on $t-1$, we can generalize the dependency structure by incorporating multiple past time steps, leading to:

$$x_{t,d} = f_d(x_{t-1,1}, \ldots, x_{t-1,D}, \ldots x_{t-L,1}, \ldots, x_{t-L,D}) + \varepsilon_t. \tag{2.3}$$

The nature of these dependencies itself can vary over time. To capture this time variation explicitly, we can express $x_{t,d}$ as a sum of functions, each focusing on a different lag:

$$x_{t,d} = f_d^{(1)}(x_{t-1,1}, \ldots, x_{t-1,D}) + \cdots + f_d^{(L)}(x_{t-L,1}, \ldots, x_{t-L,D}) + \varepsilon_t. \tag{2.4}$$

In this formulation, each function $f_d^{(i)}$ captures how the values of all D channels at time $t-i$ contribute to the evolution of channel d at time t. This significantly increases the complexity of the dependency structure, making the modeling process more intricate.

Several challenges arise from this formulation:

- **Exploding dependencies:** The number of dependencies increases with the look-back window L, leading to a rapid growth in the number of interactions that a model must learn.
- **Time-varying relationships:** The functions $f_d^{(i)}$ may themselves change over time, introducing additional non-stationarity (Zhao & Shen, 2024).
- **Uneven influence across lags:** Some channels may have stronger influences at certain lags than others, requiring models capable of dynamically weighting dependencies.

Common Assumption: Channel Independence. Despite the complexity of inter channel relationships, many works assume that each channel evolves independently. This simplification leads to an alternative formulation where each $x_{t,d}$ depends only on the past values of the same channel:

$$x_{t,d} = f_d^{(1)}(x_{t-1,d}) + \cdots + f_d^{(L)}(x_{t-L,d}) + \varepsilon_t. \tag{2.5}$$

This assumption effectively removes all cross-channel dependencies and treats each dimension separately. While such a formulation reduces model complexity, it can lead to suboptimal performance.

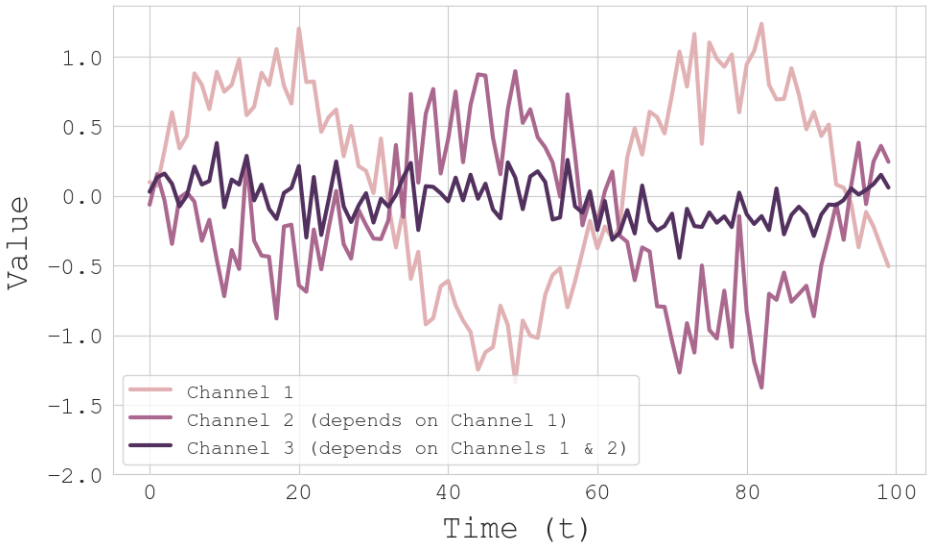


Figure 2.4: Illustration of channel interdependence in a multivariate time series. Channel 2 is partially dependent on Channel 1, while Channel 3 is influenced by both Channel 1 and Channel 2. These dependencies can evolve over time, making multivariate time series modeling more complex than the univariate case.

These factors make multivariate time series considerably more complex than their univariate counterparts. Effectively capturing both inter-channel relationships and long-range dependencies requires advanced architectures, such as recurrent networks, transformers, or graph-based models (Z. Wu, S. Pan, Long, Jiang, Chang, et al., 2020).

Time-Varying and Nonlinear Dependencies. Interdependencies between channels are often dynamic rather than static. For instance, in financial markets, the correlation between different asset prices may fluctuate based on macroeconomic conditions. Similarly, in physiological signals such as EEG, the relationship between different brain regions may vary depending on cognitive states. Traditional statistical models often assume fixed linear dependencies, but real-world multivariate time series frequently require more flexible models, such as attention-based architectures, to capture these shifting relationships (Z. Wu, S. Pan, Long, Jiang, Chang, et al., 2020).

Illustration of Channel Interdependence. Figure 2.4 provides an example of a multivariate time series where different channels exhibit interdependencies. Channel 2 depends partially on Channel 1, while Channel 3 is influenced by both Channel 1 and Channel 2. Such dependencies introduce challenges in forecasting and representation learning, as models must capture both temporal patterns within each channel and cross-channel relationships that evolve over time.

2.1.6 Applications of time series

Time series data are used in many different fields, showing how useful they are for solving real-world problems. In industry, *predictive maintenance* is one of the main applications. In this context, sensor data from machines are studied to find signs of possible future breakdowns. For example, vibration measurements from turbines or temperature readings from engines are checked regularly to spot unusual behavior. This helps plan repairs ahead of time and avoid unexpected stops. It also reduces costs and improves safety, especially in important sectors like aviation, manufacturing, and energy.

In *medical and biological data analysis*, time series are widely used to monitor physiological signals such as electrocardiograms (ECG), electroencephalograms (EEG), and blood glucose levels. For instance, in cardiology, ECG time series are analyzed to detect arrhythmias and other heart conditions. Similarly, EEG time series play an important role in diagnosing epilepsy and other neurological disorders. Wearable devices also collect health data continuously, enabling real-time monitoring of patients and the development of personalized treatments. In genomics, time series are used to observe how gene activity changes over time, offering insights into cellular processes and the effects of treatments.

In *finance and economic forecasting*, time series are indispensable tools for analyzing and predicting market trends, stock prices, and macroeconomic indicators. Models are built to capture the inherent volatility and dependencies in financial data, providing critical inputs for decision-making in investment and risk management. For example, traders rely on predictive models to forecast stock price movements, while policymakers analyze economic indicators such as inflation and unemployment rates to guide fiscal and monetary policies. Additionally, in the insurance industry, time series models are employed to assess risks, predict claims, and optimize pricing strategies.

Beyond these primary domains, time series play a pivotal role in *climate science and environmental monitoring*. Meteorologists use time series data from weather stations and satellites to forecast weather patterns, monitor climate change, and predict extreme events such as hurricanes and floods. In agriculture, time series analysis of soil moisture and crop health data helps optimize irrigation schedules and maximize yields. Furthermore, in *transportation and logistics*, time series are used to predict traffic patterns, optimize supply chain operations, and improve public transportation schedules.

In *energy management*, time series data are vital for balancing supply and demand. For example, electricity consumption data are analyzed to forecast energy needs, enabling utilities to optimize generation and reduce waste. Renewable energy sources, such as solar and wind, also rely on time series models to predict power generation based on weather conditions.

In the field of *retail and e-commerce*, time series analysis enables demand forecasting, inventory management, and sales optimization. By predicting seasonal trends and consumer behaviors, businesses can ensure product availability and enhance customer satisfaction while minimizing holding costs.

Finally, in *security and anomaly detection*, time series data are used to monitor network traffic for cybersecurity threats, detect fraudulent transactions, and identify suspicious activities. These applications leverage the temporal nature of data to recognize unusual patterns, ensuring proactive responses to potential risks.

From industrial automation to personalized medicine, and from financial markets to environmental monitoring, time series data play a central role in improving efficiency, ensuring safety, and supporting data-driven decisions. Their sequential and dynamic nature makes them especially useful for understanding and predicting complex real-world systems across many sectors.

2.2 Time Series Classification

2.2.1 Connection with the Thesis

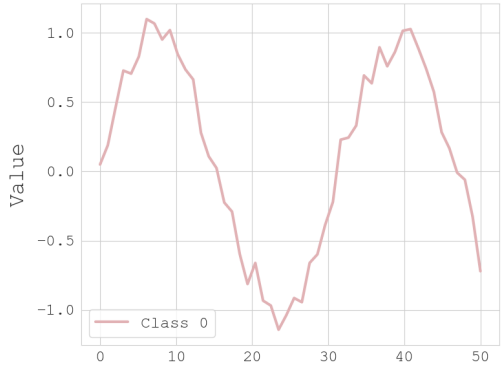
Time series classification involves assigning categorical labels to data sequences, capturing distinct temporal patterns. Traditional approaches, including distance-based and feature-based methods, have mostly addressed univariate series and rely heavily on handcrafted distance measures or carefully engineered features. In this thesis, while classification is not our primary objective, it serves as an important downstream task to validate the effectiveness of learned representations. Specifically, in Chapter 5, we approach classification from the perspective of compressing representations derived from foundation models, significantly reducing computational complexity while maintaining high accuracy. Unlike traditional time series classification methods reviewed in this section—which are primarily designed for univariate data and direct optimization for classification accuracy—our approach leverages general-purpose representations from larger models, which are then adapted to classification tasks through efficient latent space compression.

2.2.2 Introduction

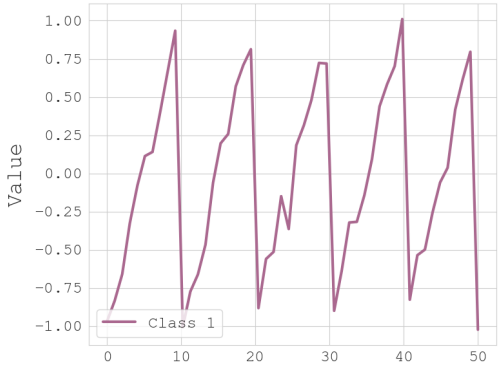
Time series classification (TSC) is the task of assigning predefined labels to sequences of data points collected over time. This problem arises in various domains such as finance, healthcare, environmental sciences, and engineering, where the goal is to categorize time-dependent patterns into meaningful classes (Ismail Fawaz, Forestier, et al., 2019; Bagnall, Lines, et al., 2017).

One of the key challenges in TSC is the variability in temporal patterns across different sequences. Figure 2.5 illustrates an example of this problem. In particular, Figure 2.5a and Figure 2.5b show representative instances from two different classes: a sinusoidal signal and a sawtooth waveform. These patterns exhibit distinct structural characteristics, which classifiers must leverage to achieve accurate predictions.

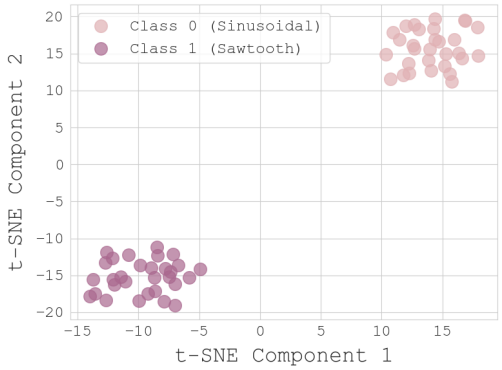
To assess the separability of these time series in a lower-dimensional space, we apply a t-SNE projection. As shown in Figure 2.5c, the two classes exhibit a good level of separation. In a more complicated scenario, overlapping regions suggest potential classification difficulties. This highlights the importance of robust feature extraction in time series classification.



(a) Example of a Class 0 time series (sinusoidal signal).



(b) Example of a Class 1 time series (sawtooth signal).



(c) t-SNE projection of the time series, highlighting class separability.

Figure 2.5: Illustration of the time series classification problem. (a) and (b) depict representative time series from two different classes, while (c) presents their distribution in a t-SNE-reduced feature space.

2.2.3 Definition of the Problem

Let $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$ be a multivariate time series of length T, where each observation $\mathbf{x}_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,D}) \in \mathbb{R}^D$ consists of D interdependent channels recorded at time step t. The task of TSC can be formulated as follows: given a training set

$$\mathcal{D} = \{(\mathbf{X}^{(i)}, y^{(i)}) \mid i = 1, \ldots, N\},$$

where $y^{(i)} \in \{1, 2, \ldots, K\}$ denotes the class label of the i-th multivariate time series, the objective is to learn a mapping function:

$$f : \mathbb{R}^{T \times D} \mapsto \{1, 2, \ldots, K\}$$

that predicts the class label of any new, previously unseen multivariate time series $\mathbf{X}$.

This problem formulation covers both univariate ($D = 1$) and multivariate ($D > 1$) time series. The challenge in TSC lies in capturing both temporal dependencies within each channel and inter-channel relationships across multiple variables (Bagnall, Lines, et al., 2017; Baydogan & Runger, 2016). Additionally, time series can vary in length, sampling rate, and noise level, requiring robust models capable of handling these variations while maintaining high classification accuracy.

2.2.4 Traditional Approaches

A wide range of approaches have been proposed for time series classification. Before the emergence of deep learning, the most commonly used techniques were broadly categorized into distance-based, feature-based, and dictionary-based methods.

Distance-based methods. Distance-based methods classify time series by computing a similarity measure between pairs of sequences and assigning a label based on the closest reference examples. The most common approach is to use a nearest neighbor classifier, where a test instance is assigned the label of its nearest neighbor in the training set. Specifically, for a given distance function $d(\mathbf{X}, \mathbf{X}')$, a test series $\mathbf{X}$ is classified as:

$$\hat{y} = \arg \min_{y^{(i)} \in \mathcal{D}} d(\mathbf{X}, \mathbf{X}^{(i)}),$$

where $\mathcal{D} = \{(\mathbf{X}^{(i)}, y^{(i)})\}_{i=1}^{N}$ is the training set and $y^{(i)}$ denotes the class label of the i-th series.

Euclidean Distance and its Limitations. The simplest similarity measure for time series is the Euclidean distance, which computes the squared differences between corresponding points in two sequences. For two univariate time series $\mathbf{X} = (x_1, \ldots, x_T)$ and $\mathbf{X}' = (x'_1, \ldots, x'_T)$ of equal length T, the Euclidean distance is defined as:

$$d_{\text{Euc}}(\mathbf{X}, \mathbf{X}') = \sqrt{\sum_{t=1}^{T} (x_t - x'_t)^2}.$$

Euclidean distance is widely used due to its simplicity and efficiency, with a computational complexity of $\mathcal{O}(T)$. However, it presents several limitations that make it less suitable for time series classification (Ding et al., 2008).

First, it assumes that time series are perfectly aligned in time. In reality, small temporal shifts between similar sequences can lead to large Euclidean distances, causing misclassifications. This sensitivity to phase variations makes Euclidean distance a poor choice for many real-world time series applications. For instance, Figure 2.6a contrasts two signals that are merely out of phase; the Euclidean approach will incorrectly penalize such shifts.

Second, while Euclidean distance is well-defined for univariate time series, its extension to the multivariate setting is less straightforward. A common approach is to compute the average Euclidean distance across all D channels of a multivariate time series:

$$d_{\text{Euc}}(\mathbf{X}, \mathbf{X}') = \frac{1}{D} \sum_{d=1}^{D} d_{\text{Euc}}(\mathbf{X}_{\cdot,d}, \mathbf{X}'_{\cdot,d}),$$

where $\mathbf{X}_{\cdot,d}$ represents the univariate time series corresponding to the d-th channel. However, this approach implicitly assumes that all channels are independent and contribute equally to the classification decision (Shokoohi-Yekta et al., 2017). This is problematic because, in many applications, such as physiological monitoring or industrial sensor data, channels exhibit complex interdependencies that are ignored by this naive averaging scheme.

As a result, Euclidean distance is generally not well-suited for multivariate time series classification, as it fails to capture cross-channel relationships and temporal misalignments. Alternative distance measures, such as Dynamic Time Warping (DTW), have been proposed to better handle these challenges (Kate, 2016).

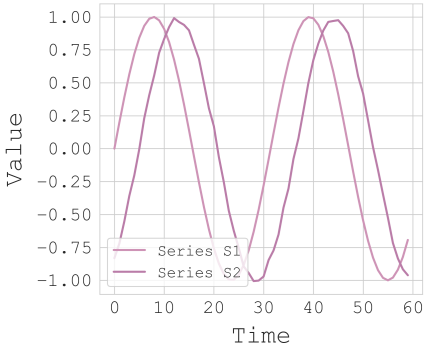
Dynamic Time Warping (DTW). DTW is a widely used measure that allows alignments between time series by introducing a warping function (Berndt & Clifford, 1994). The idea is to find an optimal alignment between two sequences by minimizing the cumulative cost of warping:

$$d_{\text{DTW}}(\mathbf{X}, \mathbf{X}') = \min_{\pi} \sum_{(t,t') \in \pi} (x_t - x'_{t'})^2,$$

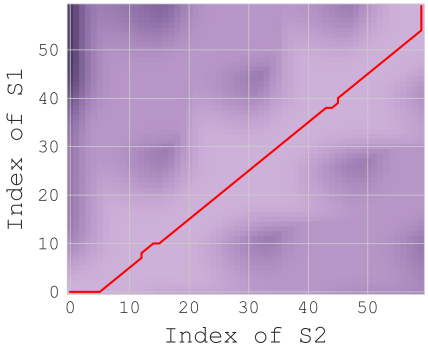
where π is a warping path that defines an alignment between $\mathbf{X}$ and $\mathbf{X}'$.

The optimal path is typically computed via dynamic programming using a *cumulative cost matrix*, denoted D. The element $D(t, t')$ represents the minimal cost of aligning the subsequences $\mathbf{X}_{1..t}$ and $\mathbf{X}'_{1..t'}$ (i.e. the first t points of $\mathbf{X}$ and the first t' points of $\mathbf{X}'$). The recurrence relation is given by:

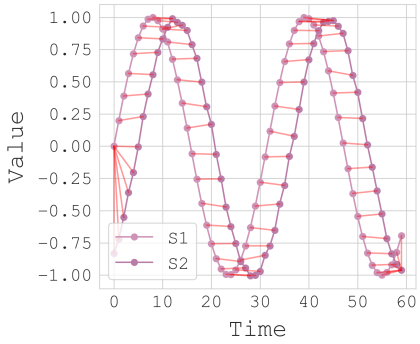
$$D(t, t') = \min\Big\{ D(t-1, t'),\ D(t, t'-1),\ D(t-1, t'-1) \Big\} \ +\ (x_t - x'_{t'})^2.$$



(a) Two signals.



(b) Warping path in DTW cost matrix.



(c) Final DTW alignment.

Figure 2.6: Comparison of Euclidean vs. DTW. (a) illustrates the original signals. (b) shows DTW cost matrix with the warping path in red and (c) the resulting point-to-point alignment, highlighting how DTW handles temporal shifts more effectively than a strict Euclidean comparison.

Once D is fully computed, the DTW distance is read off from its bottom-right corner, i.e. $d_{\text{DTW}}(\mathbf{X}, \mathbf{X}') = D(T, T')$, where T is the length of $\mathbf{X}$ and $\mathbf{X}'$ is of the same or different length. DTW is robust to phase shifts and local distortions, making it effective for time series classification. However, its quadratic complexity $\mathcal{O}(T^2)$ makes it computationally expensive for large-scale datasets (Rakthanmanon et al., 2012). Figure 2.6b illustrates the warping path computed in a DTW cost matrix, while Figure 2.6c shows how DTW realigns the two signals for a much more appropriate correspondence than Euclidean distance would yield.

Variations and Optimizations of DTW. The main issue with DTW is that its computational complexity is quadratic, i.e., $\mathcal{O}(T^2)$, where T is the length of the time series. This makes it impractical for large datasets. To address this, various improvements have been proposed to make DTW faster and more efficient while maintaining its accuracy.

- **DTW with Global Constraints:** The standard DTW algorithm allows for uncon-

strained alignments between time series, permitting any point in one sequence to be matched with any point in the other. While this flexibility ensures an optimal alignment in theory, it can also result in extreme temporal distortions, yielding unrealistic correspondences. To mitigate this issue, global constraints are introduced to confine the warping path within a predefined region, thereby enforcing a more structured alignment and reducing computational complexity. Two widely adopted constraints include:

  - *Sakoe-Chiba Band* (Sakoe & Chiba, 1978): This constraint restricts the warping path to a symmetric band around the main diagonal of the DTW cost matrix. By limiting the extent to which time steps can deviate from a direct one-to-one alignment, it prevents excessive stretching and reduces the number of admissible alignments, significantly accelerating the computation to $\mathcal{O}(TW)$, where W is the size of the band.
  - *Itakura Parallelogram* (Itakura, 1975): Unlike the fixed-width Sakoe-Chiba Band, the Itakura parallelogram adapts its shape dynamically, forming a region that expands as one moves further along the time axis, reducing the complexity to $\mathcal{O}(TlogT)$. This structure is particularly effective for speech and audio processing, where variations in speaking speed often require more flexible but still controlled warping.

  These constraints enhance computational efficiency by reducing the search space of DTW.

- **Lower Bounding Techniques:** A fundamental limitation of DTW lies in its quadratic complexity, making it computationally expensive when comparing large collections of time series. However, in many practical scenarios, it is possible to preemptively determine whether two time series are sufficiently dissimilar without computing their full DTW distance. Lower bounding techniques leverage this principle to prune unnecessary computations, thereby significantly improving efficiency. For instance, the *LB-Keogh* (Keogh & Ratanamahatana, 2005) is an approach that constructs an envelope around a reference time series, defining an upper and lower bound at each time step. If the query series deviates beyond these bounds at any position, it is guaranteed that the DTW distance exceeds a certain threshold, allowing early rejection without explicit computation. This method is particularly advantageous in nearest-neighbor search, where a large fraction of candidates can be quickly discarded. These techniques are extremely useful when searching for similar time series in large databases because they can quickly eliminate many candidates.

- **FastDTW:** (Salvador & Chan, 2007a) is an approximation method that reduces the complexity of DTW to $\mathcal{O}(T \log T)$ by adopting a multi-resolution approach. First, it begins with a *downsampling* of the time series, reducing the number of points to process. Then, DTW is applied to this simplified version, providing an initial approximate alignment at a lower computational cost. Finally, the resolution is gradually

increased, and the warping path is refined accordingly to improve the initial alignment. Thanks to this hierarchical approach, FastDTW offers an effective trade-off between speed and accuracy.

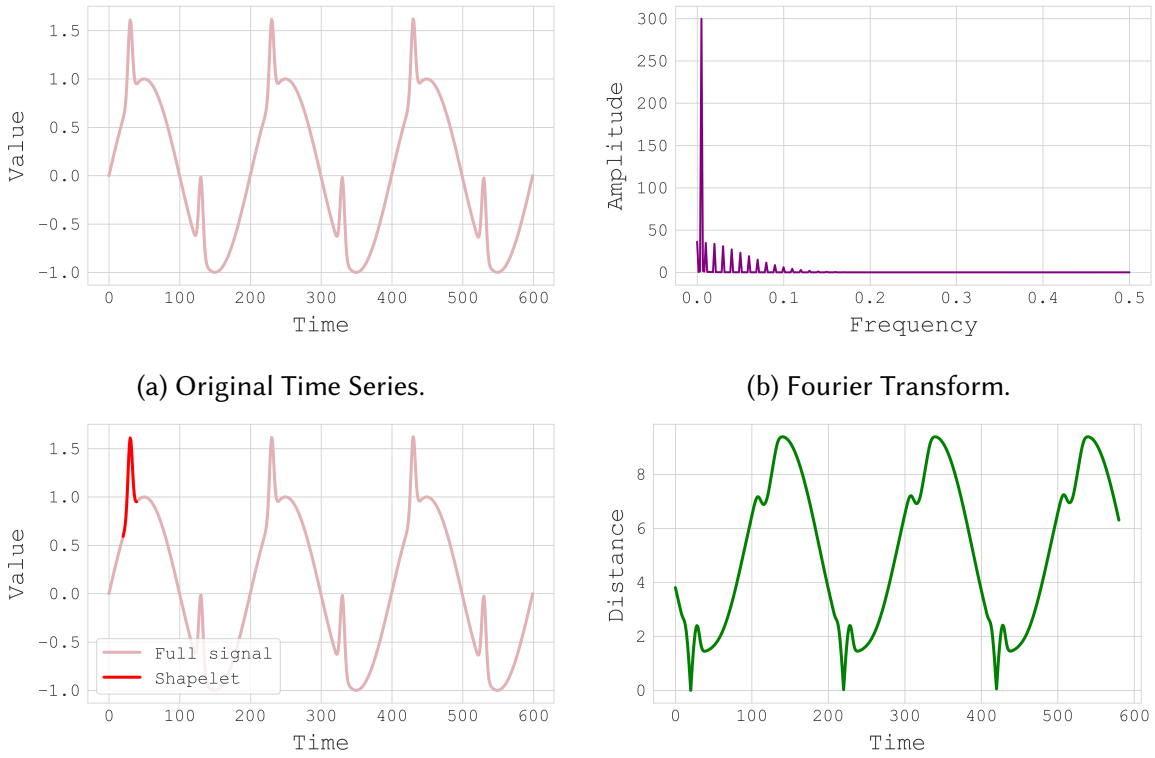
- **SoftDTW:** One of the main drawbacks of DTW is its lack of *differentiability*, which prevents its direct integration into gradient-based optimization frameworks such as deep learning. SoftDTW (Cuturi & Blondel, 2017) overcomes this limitation by introducing a differentiable relaxation of DTW. To achieve this, it replaces the standard min operator in the DTW recursion with a *soft minimum*, computed using the *log-sum-exp* function, allowing for smooth transitions between alignment costs. This reformulation enables DTW to be used as a *loss function* within machine learning models, facilitating end-to-end training. SoftDTW also enhances model robustness in tasks such as representation learning and sequence alignment in neural architectures.

These techniques are particularly beneficial in high-dimensional time series analysis, where direct DTW computations may be impractical.

Alternative Distance Measures. Other similarity measures have been developed to handle different challenges in time series classification. The Edit Distance on Real Sequences (EDR) measures similarity by counting the number of insertions, deletions, and substitutions required to align two sequences, providing robustness to noise and missing values (L. Chen et al., 2005). The Longest Common Subsequence (LCSS) computes similarity based on the length of the longest matching subsequence, which makes it particularly robust to noise and outliers (Vlachos et al., 2002). Finally, the Time Warp Edit Distance (TWED) introduces a penalty term to explicitly account for temporal differences between aligned points, thus improving robustness to temporal shifts (Marteau, 2009).

Nearest Neighbor Classification with DTW. A commonly used classifier in time series classification is the 1-NN DTW, where the class label of a test series is assigned based on its nearest neighbor under DTW distance (Bagnall, Lines, et al., 2017). Despite its simplicity, 1-NN DTW has been shown to be a strong baseline, outperforming many feature-based methods (Dau et al., 2019). However, DTW-based nearest neighbor classification suffers from several limitations. First, it has a high computational cost, as without optimizations, its complexity is $\mathcal{O}(NT^2)$, making it impractical for large datasets. Moreover, DTW does not naturally produce interpretable features, limiting its applicability in explainable models. Finally, its performance is sensitive to warping parameters, such as the warping window constraints, which require careful tuning (Lines, L. M. Davis, et al., 2015).

Feature-Based Methods. Feature-based approaches involve transforming raw time series data into a set of informative features that can be utilized by standard machine learning algorithms for classification tasks (Fulcher & Jones, 2014). This transformation aims



(a) Original Time Series.

(b) Fourier Transform.

(c) Shapelet Extraction.

(d) Shapelet Transform.

Figure 2.7: (a) The original signal has a sinusoidal baseline plus a repeated bump. (b) Fourier transform pinpoints dominant frequencies, here dominated by the low-frequency sine plus harmonics from the bump. (c) A short subsequence (shapelet) is extracted from the first occurrence of the bump. (d) Shapelet transform (distance profile) shows low-distance minima at each period where the shape recurs.

to capture the essential characteristics and underlying patterns of the time series, making them more effective and interpretable for downstream modeling.

Specifically, we illustrate statistical descriptors, frequency-domain analysis, and shapelets on a single time series. Figure 2.7a shows an example signal that combines a sinusoidal baseline with a recurrent bump. Meanwhile, the Fourier transform in Figure 2.7b identifies the dominant frequencies, including the low-frequency sine component plus smaller harmonic peaks from the bump. For shapelets, Figure 2.7c demonstrates how a short subsequence (shapelet) can be extracted from the first occurrence of the bump. The resulting shapelet transform in Figure 2.7d measures the distance between this shapelet and every subsequence in the entire signal, showing clear minima at each repetition of the bump. These techniques—statistical, spectral, and shapelet-based—can be used independently or combined to characterize a wide range of time series patterns.

- **Statistical Descriptors.** These features summarize the fundamental statistical properties of time series data, offering insights into their distribution and temporal dynamics. The mean provides a measure of the central tendency, while the variance quantifies dispersion around the mean, both foundational for understanding the overall behavior of the time series (H. Deng et al., 2013). Skewness assesses the asymmetry of the data distribution, indicating whether data are skewed towards higher or lower values, whereas kurtosis measures the "tailedness" of the distribution, revealing the presence of outliers or extreme values (H. Deng et al., 2013). Autocorrelation coefficients measure the correlation between observations at different time lags, identifying repeating patterns or periodicity within the time series (H. Deng et al., 2013). Finally, Hjorth parameters, comprising activity, mobility, and complexity, offer insights into the signal's power, frequency characteristics, and temporal structure, particularly useful for analyzing non-stationary biomedical signals like EEG and EMG (Hjorth, 1970).

- **Frequency-Domain Features.** Analyzing the frequency components of time series reveals periodic structures and oscillatory behaviors not readily apparent in the time domain. Figure 2.7b illustrates how the Fourier transform can locate dominant frequencies linked to both sinusoidal baselines and recurrent patterns. The Fourier transform decomposes the time series into sinusoidal components, identifying dominant frequencies and amplitudes, aiding in the detection of periodic patterns (Keogh & Kasetty, 2001). The wavelet transform decomposes signals at various scales, capturing frequency and temporal information effectively for non-stationary or transient signals (Addison, 2017). Additionally, the Hilbert-Huang transform, an empirical method decomposing time series into intrinsic mode functions, provides instantaneous frequency information, making it particularly suited for analyzing non-linear and non-stationary signals, such as biomedical and geophysical data (Huang et al., 1998).

- **Shapelets.** Shapelets are discriminative subsequences within time series highly indicative of class membership, capturing localized patterns for distinguishing classes. Figure 2.7c demonstrates how a shapelet can be extracted from a specific region of

the signal, while Figure 2.7d shows the resulting distance profile (shapelet transform), highlighting occurrences across the series. Shapelet discovery involves identifying subsequences maximizing class separation, subsequently used as features in classification models to capture class-specific patterns (Ye & Keogh, 2009). The shapelet transform converts original time series data into a feature space defined by distances to discovered shapelets, enabling conventional classifiers on transformed data (Hills et al., 2014). Moreover, the generalized random shapelet forest integrates shapelet-based features into a random forest framework, enhancing classification by capturing diverse discriminatory patterns (Karlsson et al., 2016).

Dictionary-Based Methods. Dictionary-based approaches represent time series as symbolic patterns (words), enabling text-like processing. These methods involve discretizing time series into symbolic representations, followed by histogram-based feature extraction. Symbolic Aggregate approXimation (SAX) converts continuous time series into discrete symbol sequences while preserving essential structural information (J. Lin, Keogh, Lonardi, et al., 2003). Bag-of-SAX-Symbols (BOSS) constructs histograms of symbolic words derived from sliding windows to capture recurrent motifs in time series data (Schäfer, 2015). SAX-VSM (Vector Space Model) extends SAX by applying a term-frequency weighting scheme, emphasizing the most informative subsequences (Senin & Malinchik, 2013).

Ensemble Methods. Ensemble classifiers improve classification performance by aggregating multiple models trained on different representations of the data. One of the most influential ensemble classifiers in time series classification is the *Collective of Transformation-Based Ensembles* (COTE), which combines models trained on diverse representations, such as transformations into different feature spaces (Lines & Bagnall, 2018). An improved variant, HIVE-COTE, integrates hierarchical voting mechanisms along with additional feature spaces, leading to state-of-the-art accuracy on benchmark datasets (Lines, S. Taylor, et al., 2018). More recently, HIVE-COTE was extended to HIVE-COTE 2.0, incorporating new components like the hierarchical ensemble framework with additional classifiers (e.g., TDE, Arsenal, DrCIF), designed to capture a wider range of complex patterns in time series data, thereby further improving classification performance (Middlehurst et al., 2021).

2.2.5 Deep Learning-Based Approaches

With the rise of deep learning, neural network-based methods have become prominent for time series classification. Their key advantage lies in learning hierarchical representations directly from raw data, avoiding manual feature engineering (Ismail Fawaz, Forestier, et al., 2019). Common deep architectures include:

Convolutional Neural Networks (CNNs). CNNs can capture local time dependencies via convolution filters, making them effective for time series classification. One notable

example is the *Fully Convolutional Network (FCN)* (Z. Wang et al., 2017), which employs multiple convolutional layers to extract spatial-temporal features. Another widely used architecture is *ResNet*, adapted for time series by incorporating residual connections to mitigate vanishing gradient issues and improve deep feature learning (K. He et al., 2016; Ismail Fawaz, Forestier, et al., 2019). More recently, *InceptionTime* (Ismail Fawaz, Lucas, et al., 2020) has extended the Inception architecture to time series data, leveraging multi-scale convolutions to enhance feature extraction across different temporal resolutions. These models have demonstrated strong performance across various time series classification benchmarks, highlighting the effectiveness of CNNs in capturing hierarchical and local dependencies in sequential data.

Recurrent Neural Networks (RNNs). RNNs are designed to handle sequence data. Variants like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) address the issues of vanishing/exploding gradients in vanilla RNNs, allowing them to capture long-term dependencies (Hochreiter & Schmidhuber, 1997a).

Transformers. Originally introduced for Natural Language Processing (NLP), Transformers rely on self-attention mechanisms. Recent work has applied Transformer-based models to time series classification and forecasting, showing competitive results (M. Liu et al., 2021). Key advantages include parallel processing and the ability to capture both short-range and long-range dependencies without recurrent operations.

2.2.6 Benchmark Datasets

Evaluating time series classification algorithms typically involves standardized repositories, metrics, and protocols. This section provides an overview of popular benchmark datasets and commonly used performance metrics.

UCR/UEA Time Series Classification Archive. The *UCR/UEA Archive* is one of the most comprehensive and widely used repositories dedicated to benchmarking time series classification algorithms. Initially developed at the University of California, Riverside (UCR) for univariate datasets and later extended by the University of East Anglia (UEA) to include multivariate datasets, it currently hosts over 150 diverse datasets. These datasets cover various domains, including image outlines (e.g., BeetleFly, FaceAll, DiatomSizeReduction), sensor readings (e.g., InsectWingbeat, Car, Cricket), and biomedical signals (e.g., ECG200, ECG5000, EEG Eye State). The archive's datasets vary significantly in length (from tens to thousands of time steps), number of classes (ranging from binary to multi-class problems with dozens of classes), and dimensionality, featuring a large collection of univariate datasets (UCR) alongside an increasing number of multivariate datasets maintained within the UEA archive (Dau et al., 2019; Bagnall, Dau, et al., 2018a). The archive

provides train/test splits for each dataset to standardize comparisons and ensure reproducibility, and researchers often measure performance using accuracy on these predefined splits (Dau et al., 2019). Some tasks, especially within the multivariate datasets, may also require domain-specific preprocessing or normalization.

Additional Benchmarks. Beyond UCR/UEA, other notable benchmarks include:

- **Physionet Collections**: Contain ECG and other physiological signals, often used for arrhythmia detection (Goldberger et al., 2000).
- **PLAsTiCC Challenge Dataset** (Hložek et al., 2020): Contains simulated astronomical time-series data (light curves) from 14 different types of astronomical objects, designed for classification tasks in astrophysics.

2.2.7 Evaluation Metrics for Time Series Classification

Several evaluation metrics can be employed, each offering different insights:

- **Accuracy**:
$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of samples}}.$$
This is the most common metric, reflecting the overall proportion of correctly classified instances. However, it can be misleading in imbalanced-class scenarios, where one class dominates.
- **Precision**:
$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{TP} + \text{False Positives (FP)}}.$$
Precision answers: "Of all samples predicted as positive, how many are truly positive?"
- **Recall** (*a.k.a. Sensitivity*):
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{False Negatives (FN)}}.$$
Recall answers: "Of all positive samples, how many did we correctly identify?"
- **F1-Score**:
$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$
This is the harmonic mean of precision and recall. F1-Score is particularly helpful in handling class-imbalance, as it balances both false positives and false negatives.

- **Balanced Accuracy**:

$$\text{Balanced Accuracy} = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i},$$

  where C is the number of classes. Balanced accuracy averages the recall for each class and is recommended when class imbalance is severe.

- **ROC Curve & AUC (Area Under the ROC Curve)**: The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The *AUC* summarizes the ROC curve as a single number between 0 and 1, indicating the classifier's overall ability to discriminate between classes. High AUC means high separability (Fawcett, 2006).

- **Precision-Recall Curve & Average Precision (AP)**: More informative than ROC in highly imbalanced datasets, the precision-recall curve plots precision vs. recall. The *AP* (average precision) is the area under this curve, providing insight into performance under different threshold settings (Saito & Rehmsmeier, 2015).

Overall, the combination of standardized benchmarks (UCR/UEA) and a diverse set of metrics (accuracy, F1, ROC-AUC, etc.) provides a solid foundation for evaluating time series classification algorithms in both academic and industrial contexts.

2.2.8 State of the Art in Time Series Classification

Recent advances in time series classification (TSC) have led to a three-way contest among fast random-kernel methods, advanced ensemble approaches, and deep learning architectures. Below is an overview that integrates recent developments and highlights key differences between univariate and multivariate TSC.

Univariate TSC. On standard univariate datasets (e.g., the UCR Archive), three primary families have emerged:

- **ROCKET-based Models:** Methods like ROCKET, Multi-ROCKET, and MiniRocket leverage large numbers of random convolutional kernels to rapidly transform time series data. They achieve state-of-the-art performance with training times measured in seconds or minutes, exhibiting linear complexity in both series length and the number of training examples (Dempster et al., 2020a; Tan et al., 2022).

- **Ensemble Methods:** Approaches such as HIVE-COTE (and its successor HIVE-COTE 2.0) combine heterogeneous classifiers—including shapelet, dictionary, interval, and similarity-based methods—to maximize accuracy. Despite high computational demands, these ensembles consistently achieve top-ranked accuracy on benchmark tasks (Middlehurst et al., 2021; Shifaz et al., 2020).

- **Deep Convolutional Architectures:** Models such as InceptionTime utilize multiple convolutional modules, often with residual connections, to learn complex representations directly from raw time series data. These methods frequently outperform traditional baselines, especially on larger datasets (Ismail Fawaz, Lucas, et al., 2020).

In practice, accuracy differences among these methods are often marginal, suggesting that dataset-specific characteristics (e.g., series length, noise, pattern complexity, and interpretability needs) can determine the best-suited method.

Multivariate TSC. Multivariate time series classification introduces additional challenges due to the inherent complexity and high dimensionality of multi-channel data. Key approaches in this context include:

- **Multivariate Ensembles:** Ensemble methods specifically tailored for multivariate data, such as WEASEL+MUSE (Schäfer & Leser, 2017), DrCIF (Middlehurst et al., 2021), and adapted versions of HIVE-COTE, handle multi-channel inputs effectively by incorporating dimension-specific weighting or channel-selection mechanisms. However, training times often become prohibitive as dimensionality increases.
- **Transformer-based Models:** Recently, Transformer architectures leveraging self-attention and cross-attention mechanisms (e.g. TimesNet) have shown state-of-the-art performance in multivariate classification tasks. These models excel at capturing long-range dependencies across channels but come with significant computational resource demands, especially GPU memory (Zerveas et al., 2021; Yixuan Nie et al., 2023).
- **CNN/RNN Hybrids:** Architectures combining convolutional and recurrent networks (e.g., LSTM-FCN) capture correlations among channels efficiently, performing particularly well when multivariate channels are highly correlated. Nevertheless, these models may underfit in scenarios with heterogeneous channels or limited training data, since capturing complex interactions typically requires larger datasets (Karim et al., 2019; Seyfi et al., 2022).

The transition from univariate to multivariate TSC is not merely an increase in dimensionality; it demands reconsidering model architectures to effectively fuse information across channels and manage computational overhead. Additionally, interpretability considerations may favor methods such as shapelet or dictionary-based classifiers, which explicitly identify discriminative patterns and facilitate explanations of model predictions (Ye & Keogh, 2009; Schäfer, 2015).

In summary, the state-of-the-art in TSC reflects a nuanced trade-off among computational efficiency, predictive accuracy, and interpretability. Univariate TSC benefits from well-tuned random-kernel (ROCKET variants) and ensemble methods (HIVE-COTE), while multivariate TSC requires specialized adaptations, including Transformers, to address increased complexity. Model selection thus depends heavily on dataset characteristics, computational resources, and interpretability requirements.

2.3 Time Series Forecasting

2.3.1 Connection with the Thesis

Multivariate time series forecasting involves analyzing historical observations to predict future values, a task complicated by temporal dependencies, channel interactions, and non-stationarity. This thesis directly addresses these challenges, as forecasting constitutes one of our core contributions. In Chapter 3, we introduce SAMformer, which departs from traditional transformer-based methods by incorporating sharpness-aware minimization and channel-wise attention to enhance generalization, unlike existing approaches that primarily rely on decomposition techniques or assume independence between channels. Moreover, in Chapter 4, we propose a multi-task regularization framework specifically designed to improve forecasting performance of models that assume channel independence. By integrating this regularization into the learning objective, we explicitly enforce shared representations across channels, significantly boosting predictive accuracy and enhancing interpretability. Thus, while we address a similar forecasting problem as prior work, our methods differ substantially by optimizing across channels and refining the optimization landscape to achieve more robust and interpretable solutions.

2.3.2 Introduction

Time series forecasting is the task of predicting future values of a time-dependent sequence based on its past observations and other exogenous variables. It is a critical problem in many fields, including finance, supply chain management, energy, and healthcare. This section reviews the fundamental definitions, traditional approaches, machine learning methods, and recent deep learning models for time series forecasting.

2.3.3 Definition and Problem Formulation

Time series forecasting aims to predict future observations based on historical data, generally under constraints of non-stationarity, domain shifts, and variable horizon lengths (George E. P. Box et al., 2015; Hyndman & Athanasopoulos, 2008). Let $\mathbf{X}_t \in \mathbb{R}^d$ be the vector of observations at time t, and consider a univariate or multivariate time series $\{\mathbf{X}_t\}_{t=1}^T$. The forecasting task is to predict $\{\mathbf{X}_{T+1}, \ldots, \mathbf{X}_{T+H}\}$ for a horizon $H > 0$, given past observations $\{\mathbf{X}_1, \ldots, \mathbf{X}_T\}$ and possibly n_{cov} external covariates $\{\mathbf{Z}_t\}$. We define the forecasting function as a mapping:

$$f : \mathbb{R}^{T \times D} \times \mathbb{R}^{T \times n_{\text{cov}}} \longrightarrow \mathbb{R}^{H \times D},$$

where each input sequence is given by $\{\mathbf{X}_t\}_{t=1}^T$ with $\mathbf{X}_t \in \mathbb{R}^D$ and $\{\mathbf{Z}_t\}_{t=1}^T$ with $\mathbf{Z}_t \in \mathbb{R}^{n_{\text{cov}}}$. The function f then outputs a forecast

$$\{\hat{\mathbf{X}}_{T+1}, \ldots, \hat{\mathbf{X}}_{T+H}\} \in \mathbb{R}^{H \times D},$$

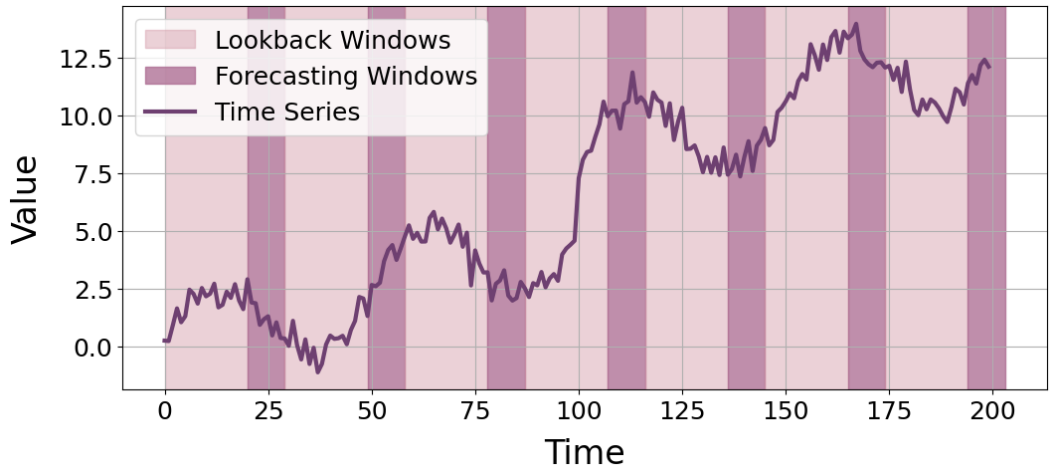


Figure 2.8: Illustration of a forecasting problem. The time series is divided into looback and forecasting pairs.

representing the predicted values over the future horizon of length H.

Short- vs. Long-Horizon Forecasting. Forecasting horizons can vary significantly. Short-term forecasting (often defined as $H < 96$) generally focuses on intervals from a few hours to a few days, making it suitable for operational decisions (e.g., traffic, energy load). Long-term forecasting ($H \geq 96$), by contrast, may extend over weeks, months, or even years, necessitating robust handling of seasonality, trends, and evolving external conditions. Recent empirical findings indicate that Transformers may outperform simpler baselines for $H < 96$, but for longer horizons ($H \geq 96$), linear or more specialized models can surpass Transformers (Zeng, M. Chen, et al., 2023).

Domain Shift and Non-Stationarity. In practice, distributional changes (domain shifts) between training, validation, and test sets complicate forecasting. Seasonal patterns, abrupt events, or sensor drift can introduce non-stationarities that degrade model performance over time. While some models (e.g., adaptive or recurrent neural networks) partially address such shifts, consistent re-training or model adaptation is often required in real-world deployments (George E. P. Box et al., 2015).

Relation to Time Series Classification. TSC typically deals with shorter series like seconds of accelerometer data in UCR/UEA archives (Bagnall, Lines, et al., 2017; Dau et al., 2019), focusing on assigning labels based on shape or local patterns. Forecasting, by contrast, often involves substantially longer sequences —sometimes spanning years— and emphasizes predicting future values rather than categorizing entire sequences. Consequently, TSC datasets may be seconds or minutes in duration, whereas forecasting problems can easily range from days to years of data. This scale difference strongly influences model choice: short TSC tasks may favor shapelet-based or CNN approaches, while long-horizon

forecasting challenges highlight issues of trend, seasonality, and domain shift.

Challenges in Real-World Settings. In practical scenarios, multivariate time series forecasting often confronts missing data and outliers stemming from sensor failures, extreme events, or irregular sampling, thereby necessitating robust preprocessing or imputation. Moreover, non-linear and often intricate interdependencies across multiple channels demand more advanced architectures (e.g., attention mechanisms, graph-based approaches). Lastly, long-term deployments require continual model maintenance, as new patterns may emerge and call for periodic updates or re-training.

In the following, we briefly survey classical statistical approaches before examining advanced neural architectures. While historical techniques like ARIMA or exponential smoothing offer interpretability and strong performance in stable conditions, they can be outperformed by deep or hybrid methods, particularly in high-dimensional or highly non-stationary contexts (Hyndman & Athanasopoulos, 2008).

2.3.4 Traditional Statistical Methods

Traditional forecasting models prioritize interpretability and simplicity, relying on parsimonious structures to represent temporal patterns effectively. Classical autoregressive (AR) and moving average (MA) models assume linear relationships between past observations or noise terms, forming the foundations of time series modeling. ARIMA (George E. P. Box et al., 2015) extends these approaches by introducing differencing operations to handle trends and achieve stationarity, effectively capturing linear autocorrelations. SARIMA further generalizes ARIMA by explicitly modeling seasonal components, making it particularly suitable for data exhibiting periodic behavior.

Exponential smoothing methods (ETS) offer another widely-used class of forecasting tools. They generate forecasts by computing weighted averages of past observations with exponentially decaying weights, naturally emphasizing more recent data points (Hyndman & Athanasopoulos, 2008; Gardner, 2006). The comprehensive ETS framework developed by Hyndman, Koehler, et al. 2002 encompasses various configurations of error, trend (additive or multiplicative), and seasonality, often excelling in stable univariate settings with clear seasonal patterns.

Several advanced extensions have also emerged to address more complex seasonal structures and nonlinear patterns. TBATS (Livera et al., 2011) expands the ETS approach by integrating trigonometric functions to represent intricate or multiple seasonalities and applying Box-Cox transformations to manage nonlinearities and heteroskedasticity. Prophet (S. J. Taylor & Letham, 2018), developed by Meta (formerly Facebook), employs a decomposable framework, combining piecewise linear growth trends with custom seasonal and holiday effects. Its strength lies in interpretability and the explicit integration of domain-specific knowledge, making it appealing for practical forecasting applications.

Despite their simplicity and widespread use, traditional methods can struggle with

highly nonlinear relationships, significant domain shifts, or multivariate complexities, highlighting the need for more flexible forecasting approaches.

2.3.5 Machine Learning Approaches

Machine learning (ML) techniques have emerged as flexible alternatives to traditional statistical models, particularly effective at modeling nonlinear relationships, complex dependencies, and multivariate interactions (Ahmed et al., 2010; Bontempi et al., 2012). Though sometimes sacrificing interpretability, ML models often outperform classical approaches in scenarios with large or high-frequency datasets.

Tree-based methods such as Random Forests (RF) and gradient boosting frameworks (XGBoost, LightGBM, CatBoost) have become standard tools for forecasting. RF models (Breiman, 2001) leverage ensembles of decision trees trained on random subsets of data and features, reducing overfitting and capturing nonlinearities effectively. Gradient boosting methods iteratively construct ensembles by fitting residuals, often achieving superior predictive performance on extensive datasets, albeit at higher computational costs (T. Chen & Guestrin, 2016; Ke et al., 2017; Dorogush et al., 2018). Other regression approaches, such as Support Vector Regression (SVR) (Smola & Schölkopf, 2004) and Multi-Layer Perceptrons (MLP) (G. Zhang et al., 1998), can similarly achieve high accuracy, provided carefully engineered features and hyperparameter tuning.

Crucially, ML methods typically rely on feature engineering to uncover informative temporal patterns. Techniques include constructing lagged variables, applying differencing to remove trends, incorporating calendar-based features (day-of-week, month-of-year, or special events), and using Fourier terms (Harvey, 1993) to approximate complex seasonal patterns. Although powerful, extensive feature engineering can be time-consuming and difficult to maintain.

Ensemble strategies can further enhance forecasting performance by combining multiple methods to leverage their complementary strengths. Stacking or blending approaches train meta-learners on top of diverse base models (e.g., ARIMA combined with gradient boosting) to capture a broader range of patterns (Wolpert, 1992; Makridakis, Spiliotis, et al., 2020). Hybrid statistical–ML models, such as combining exponential smoothing residuals with neural networks or random forests, similarly capitalize on both linear and nonlinear structures within the data (G. Zhang, 2003).

Ultimately, the choice of forecasting method must align closely with dataset characteristics, resource availability, interpretability requirements, and forecasting horizon considerations, as each method offers distinct advantages and trade-offs.

2.3.6 Deep Learning Methods

Deep learning models have recently emerged as powerful approaches for time series forecasting, capable of capturing intricate long-range temporal dependencies, complex seasonal patterns, and interactions among multivariate channels. Unlike traditional statistical or tree-based methods, deep architectures learn useful representations directly from raw data, significantly reducing the need for explicit feature engineering. However, their predictive power often comes at the cost of reduced interpretability and increased computational requirements (Casolaro et al., 2023).

Recurrent Neural Networks (RNNs). Recurrent neural networks, particularly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), have been widely adopted due to their capacity to model sequential dependencies through recurrent updates. LSTMs (Hochreiter & Schmidhuber, 1997b) leverage gating mechanisms specifically designed to mitigate issues like vanishing gradients, enabling them to effectively capture long-term temporal correlations. GRUs (Cho et al., 2014) simplify these gating structures, reducing parameter complexity while often achieving comparable forecasting accuracy. Extensions of recurrent architectures, such as DeepAR (Salinas et al., 2020a), adopt probabilistic frameworks for forecasting distributions of future values, proving particularly effective when handling multiple correlated time series simultaneously.

Convolutional Neural Networks (CNNs). Convolution-based methods offer computational advantages over recurrent models due to their inherently parallelizable architectures. Temporal Convolutional Networks (TCN) (S. Bai et al., 2018), for example, exploit dilated convolutions and residual connections to achieve effective modeling of long-range temporal dependencies without explicit recurrent structures. Similarly, WaveNet-inspired architectures (Oord et al., 2016), originally developed for audio signal processing, have demonstrated strong performance in forecasting tasks by employing stacked dilated convolutions to efficiently model both local and global temporal features.

Other Models. Hybrid models, such as LSTNet (Guokun Lai et al., 2018a), further capitalize on the strengths of CNNs and RNNs, using convolutional layers for short-term pattern detection followed by recurrent layers to capture longer-term dependencies. Recent deep learning architectures also include graph neural networks (GNNs), notably useful when explicit relationships among variables, such as spatial or topological structures in sensor networks, can be leveraged to enhance predictive accuracy (Z. Wu, S. Pan, Long, Jiang, C. Zhang, et al., 2020).

Despite their impressive capabilities, deep learning approaches typically require large datasets and substantial computational resources. Moreover, they may demand extensive hyperparameter tuning to achieve optimal performance and interpretability requirements in applied forecasting scenarios.

Transformers and Attention-Based Models. Transformer architectures (Vaswani et al., 2017) have recently gained significant attention in time series forecasting due to their powerful self-attention mechanisms, capable of effectively capturing long-range temporal dependencies. Unlike traditional recurrent neural networks (RNNs), transformers handle sequences through parallel computations, enabling better scalability and efficiency for forecasting tasks involving long sequences. However, the standard transformer suffers from quadratic computational and memory complexity, $\mathcal{O}(T^2 d)$, where T denotes sequence length and d is the embedding dimension. This limitation becomes particularly restrictive for very long time series, motivating several adaptations specifically designed to mitigate these computational constraints.

Efficient Transformers for Long-Horizon Forecasting. To address the quadratic complexity of standard transformers, various specialized architectures have emerged:

- **Informer** (H. Zhou et al., 2021) proposes *probSparse attention*, a sparsified attention mechanism reducing complexity to $O(T \log T)$, coupled with a distillation operation to remove redundant temporal information.
- **Autoformer** (H. Wu et al., 2021) introduces a *series decomposition block* explicitly modeling trend and seasonality, replacing conventional attention with an *auto-correlation mechanism* to select relevant temporal lags efficiently.
- **FEDformer** (T. Zhou, Ma, et al., 2022) employs a frequency-domain attention scheme based on Fourier and Wavelet transforms, capturing dependencies in both time and frequency domains and significantly reducing computational load.
- **PatchTST** (Yuqing Nie et al., 2023), inspired by vision transformers, divides long time series into patches to enhance the inductive bias and effectiveness of self-attention mechanisms, leading to improved performance on extended forecasting horizons.
- **iTransformer** (Yong Liu et al., 2024) utilizes *instance-based attention* to dynamically adapt attention weights according to local temporal patterns rather than relying solely on static positional encodings.
- Additional efficient architectures include **Reformer** (Kitaev et al., 2020), which applies locality-sensitive hashing (LSH) for sublinear attention approximation, and **LogTrans** (Shiyang Li et al., 2019), employing log-sparse attention that prioritizes local and distant interactions.

Challenges and Limitations of Transformers. Despite their effectiveness, transformer-based models are not without shortcomings. Even optimized variants often require substantial computational resources, limiting their applicability in resource-constrained environments. Furthermore, recent research (Zeng, Yan, et al., 2023) highlights that transformers can underperform simpler linear or MLP-based models in extremely long forecasting horizons ($H \geq 96$), questioning their suitability for certain long-term forecasting tasks.

Additionally, standard transformers often suffer from training instability and convergence to sharp local minima, particularly when dealing with smaller datasets or limited data availability (Dong et al., 2021; L. Liu et al., 2020). While these issues have been actively addressed in computer vision and NLP through approaches like sharpness-aware minimization (SAM) (Foret et al., 2021), similar solutions remain under-explored within the time series community.

MLP-based Alternatives: Mixer Models. To circumvent complexities associated with self-attention mechanisms, recent work has proposed mixer-based architectures inspired by the original MLP-Mixer concept from computer vision (Tolstikhin et al., 2021). Specifically, TSMixer (Yujie Liu et al., 2023) has emerged as a state-of-the-art approach for long-horizon forecasting. TSMixer uses fully-connected MLP layers for both token mixing (across time steps) and channel mixing (across variables), eliminating attention mechanisms entirely and thus achieving linear computational complexity (Td). Empirical evidence demonstrates that TSMixer can outperform transformer models for very long horizons, largely due to its favorable inductive bias for temporal extrapolation (Yujie Liu et al., 2023).

Limitations of Mixer-based Models. Despite their computational advantages, mixer-based models also have limitations. By foregoing attention entirely, they may fail to adequately capture intricate, non-local temporal dependencies or complex cross-variable interactions, especially in chaotic or highly nonlinear scenarios. This trade-off can potentially reduce predictive performance in tasks demanding richer contextual modeling or deeper representations of temporal dynamics.

Summary of Trade-offs. In summary, transformer architectures provide powerful mechanisms for capturing long-range dependencies in time series, yet face significant challenges related to computational complexity, training instability, and limited performance in ultra-long forecasting horizons. Mixer-based models offer attractive alternatives with linear computational complexity and better inductive biases for extrapolation, though they may struggle with complex dependency modeling. Future advances in time series forecasting could benefit from combining the efficiency of mixers with the expressive power and optimized training strategies of transformers.

Hybrid and Advanced Architectures. Hybrid models that combine CNNs, RNNs, and Transformers exploit complementary strengths of different neural architectures to achieve superior forecasting performance. For instance, N-BEATS (Oreshkin et al., 2020) employs fully connected neural networks organized into specialized blocks, using a hierarchical structure to iteratively decompose time series into interpretable trend and seasonal components, achieving competitive performance on long-horizon forecasting tasks. Hybrid CNN-RNN models such as LSTNet (Guokun Lai et al., 2018a) integrate convolutional layers, which effectively capture local short-term patterns, with recurrent units for modeling longer-term temporal dependencies. Additionally, Graph Neural Networks (GNNs) have

emerged as powerful alternatives, particularly for multivariate forecasting tasks, by explicitly leveraging inter-channel correlations structured as graphs (Z. Wu, S. Pan, Long, Jiang, Chang, et al., 2020). While advanced deep learning models can achieve strong performance, they typically require significant computational resources and careful hyperparameter tuning. For smaller, stable, or resource-constrained forecasting tasks, simpler statistical models or tree-based ensembles may be more suitable.

2.3.7 Benchmark Datasets

Standardized benchmark datasets are key for evaluating time series forecasting models under realistic conditions. These datasets vary in terms of length, frequency, domain, and complexity, influencing the suitability of different forecasting methods. Below, we describe key datasets and benchmarks commonly used in the forecasting literature.

M-Competitions. The M-Competitions have significantly contributed in advancing the state-of-the-art in forecasting by systematically comparing statistical, machine learning, and hybrid approaches on large-scale real-world datasets. These competitions include:

- **M3** (Makridakis & Hibon, 2000): 3,003 time series covering microeconomic, macroeconomic, demographic, and industry-related data, with forecasting horizons ranging from 6 to 18 steps.
- **M4** (Makridakis, Spiliotis, et al., 2018): 100,000 time series from a variety of domains (finance, demographics, energy, etc.), spanning different frequencies (hourly, daily, weekly, monthly, quarterly, yearly).
- **M5** (Makridakis, Spiliotis, et al., 2020): Forecasting product sales at Walmart, with hierarchical demand structures and external factors such as promotions and price elasticity.

These competitions have highlighted the strengths of hybrid models combining statistical and machine learning approaches, as well as the importance of probabilistic forecasting.

Monash Time Series Forecasting Archive. The Monash Time Series Forecasting Repository (Ruwan Godahewa et al., 2021) is one of the most comprehensive collections of publicly available forecasting datasets. It includes:

- **Tourism** (Athanasopoulos et al., 2011): 1,311 time series related to Australian tourism, with monthly, quarterly, and yearly granularities.
- **Weather** (Ruwan Godahewa et al., 2021): Meteorological records across different countries, covering variables such as temperature, precipitation, and wind speed.

- **Traffic** (R. Yu et al., 2017): Road traffic sensor data collected at different locations, typically sampled at 15-minute or hourly intervals.
- **Electricity** (Dua & Graff, 2017): Records of power consumption at multiple client sites, often used to evaluate energy load forecasting models.

UCI and UEA Time Series Archives. The UCI Machine Learning Repository (Dua & Graff, 2017) and the UCR/UEA Time Series Classification Archive (Bagnall, Lines, et al., 2017; Dau et al., 2019) are primarily designed for time series classification but have also been adapted for forecasting tasks. They include datasets such as:

- **StarLightCurves** (Dau et al., 2019): Light intensity variations from astronomical observations.
- **Handwriting and Motion Capture** (Bagnall, Lines, et al., 2017): Time series derived from accelerometer and gyroscope data, useful for human activity recognition.

These datasets are often short, unlike forecasting datasets which typically span thousands of points.

Competitions and Industry Benchmarks. Several forecasting challenges have been launched by major tech companies and research communities:

- **Kaggle Web Traffic Forecasting** (Google, 2017): Time series of daily web traffic to Wikipedia pages, with thousands of noisy, sparse, and seasonal series.
- **Amazon Demand Forecasting Challenge** (Salinas et al., 2020a): Forecasting product demand at different aggregation levels, using historical sales data with exogenous variables.
- **Walmart Sales Forecasting** (Makridakis, Spiliotis, et al., 2020): Retail demand forecasting across thousands of stores and SKUs, incorporating promotions and holidays.

Dataset Characteristics and Challenges. The datasets used in forecasting vary widely in structure and characteristics:

- **Length**: Forecasting datasets often contain much longer time series (up to decades of historical data) compared to time series classification datasets, which typically have a few hundred observations.
- **Frequency**: Some datasets use high-frequency data (e.g., electricity, traffic), while others operate at much coarser time scales (e.g., yearly macroeconomic indicators).
- **Domain Shift**: Many forecasting datasets exhibit domain shifts between training and test sets due to changing economic, seasonal, or external factors.

Dataset	# Time Series	Length	Frequency
M3	3,003	Varies	Monthly, Quarterly, Yearly
M4	100,000	Varies	Hourly to Yearly
M5	42,840	∼2,000	Daily
Tourism	1,311	Varies	Monthly, Quarterly, Yearly
Traffic	17,000+	15k+	15-min, Hourly
Electricity	3,700+	26,304	Hourly
Wikipedia Web Traffic	145,000	∼550	Daily
Amazon Demand Forecast	10,000+	2,000+	Daily

Table 2.1: Comparison of commonly used forecasting datasets in terms of size, length, and frequency.

2.3.8 Evaluation Metrics for TIme Series Forecasting

The evaluation of forecasting models depends on carefully chosen error metrics, which must account for aspects such as scale dependence, sensitivity to outliers, interpretability, and applicability to probabilistic forecasts. Below, we present commonly used metrics, along with their strengths and weaknesses.

Magnitude-Dependent Metrics. These metrics evaluate absolute errors and are sensitive to the scale of the target variable.

- **Mean Absolute Error (MAE)**: Measures the average absolute difference between predictions and actual values:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i| \,.$$

  Easy to interpret (same unit as the target) and less sensitive to large errors compared to squared-error metrics, but penalizes over- and underestimates equally, which may not be suitable for all applications. It is still somewhat affected by outliers, though less so than RMSE.

- **Mean Squared Error (MSE)**: Penalizes larger errors more strongly due to the squaring term:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2.$$

  Helpful when large deviations need to be heavily penalized, and it is differentiable, which aids model optimization. However, it is more sensitive to outliers than MAE.

Scale-Independent Metrics. These metrics are useful when comparing models across datasets with different scales.

- **Mean Absolute Percentage Error (MAPE)**: Measures the error as a percentage of the actual value:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{|y_i|} \times 100.$$

  Expresses errors in percentage, making it easier to compare across different datasets and well-suited to fields like finance. However, it becomes unstable when target values are close to zero and can disproportionately penalize small denominators.

- **Symmetric Mean Absolute Percentage Error (sMAPE)**: A variant designed to mitigate issues with small values:

$$\text{sMAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{\frac{1}{2}\left(|y_i| + |\hat{y}_i|\right)} \times 100.$$

  Partially addresses the zero-value problem by balancing over- and underestimates, yet it remains sensitive to very small denominators and can behave unintuitively when both y_i and $\hat{y}_i$ are very low.

Probabilistic Forecasting Metrics. For models that produce probability distributions rather than point estimates, specific metrics are required.

- **Continuous Ranked Probability Score (CRPS)**: Evaluates the accuracy of the full predictive distribution, a widely adopted metric in probabilistic forecasting:

$$\text{CRPS}(F, y) = \int_{-\infty}^{+\infty}\Big(F(z) - \mathbb{I}\{y \leq z\}\Big)^2\, dz,$$

  where $F(z)$ is the cumulative distribution function of the forecast and $\mathbb{I}\{y \leq z\}$ is the indicator function of the observation y. CRPS is particularly advantageous over quantile loss as it does not require computing multiple quantiles to form prediction intervals.

Choosing the Right Metric. The appropriate metric depends on the specific needs of the forecasting task. For scenarios requiring clear interpretability, MAE is advantageous because it shares the same units as the target. When large deviations need to be heavily penalized, RMSE is preferable, although it can overemphasize outliers. MAPE and sMAPE are well-suited for comparisons across different datasets but must be handled carefully when target values approach zero. Overall, multiple metrics are often used together to get a comprehensive evaluation of model performance.

2.3.9 State of the Art in Time Series Forecasting

Recent advances in time series forecasting have been driven by deep learning models, particularly transformer-based architectures. While these models outperform classical approaches on short to medium forecasting horizons, they exhibit performance degradation for long-term forecasting. This section analyzes key trends, empirical findings, and limitations of state-of-the-art methods.

Short-Term vs. Long-Term Forecasting: A Performance Gap. The forecasting horizon H is a key factor in model performance. Empirical studies (Zeng, Shiyang Li, et al., 2023; H. Wu et al., 2021) indicate a clear divide:

- **Short-term forecasting ($H < 96$)**: Transformer-based models achieve superior accuracy by capturing non-linear temporal dependencies (Yuqi Nie et al., 2022; Yong Liu et al., 2024).
- **Long-term forecasting ($H \geq 96$)**: Linear models surpass deep learning models. Their simpler architectures reduce overfitting, an issue in Transformer-based forecasting (Zeng, Shiyang Li, et al., 2023; Christopher Challu et al., 2022).

Benchmarking Performance on Standard Datasets. To systematically compare forecasting models, research relies on standardized benchmarks:

Dataset	# Channels	Frequency
ETTh1 / ETTh2	7	1 hour
ETTm1 / ETTm2	7	15 min
Traffic	862	1 hour
Electricity	321	1 hour
Weather	21	10 min
Exchange-Rate	8	1 day

Table 2.2: Common benchmarks for evaluating time series forecasting models.

We observe that a notable shift between the training and test distributions affects all models. Deep learning methods, however, prove generally more sensitive to previously unseen patterns and unexpected variations in the data.

Current Research Challenges and Open Questions. Current research in forecasting continues to face several unresolved challenges. First, scalability is a significant concern: self-attention can exhibit quadratic complexity in sequence length, resulting in high computational costs for high-dimensional multivariate series (H. Zhou et al., 2021). Second,

while traditional statistical models present clear and interpretable parameters, deep learning approaches often remain opaque, necessitating post-hoc methods to explain model decisions. Finally, domain adaptation remains a challenge, since real-world distribution shifts often prevent models from generalizing effectively. Potential solutions include hybrid architectures combining statistical decomposition and deep feature extraction, as well as memory-efficient self-attention mechanisms for long-term forecasting (H. Wu et al., 2021).

2.3.10 Conclusion

Although recent studies highlight good outcomes for transformer-based models in short-term prediction, a key limitation emerges when forecasting horizons grow beyond $H \geq 96$. In these scenarios, linear methods continue to lead the field. Moving forward, three priorities stand out: first, boosting transformers' efficiency to reduce high computational overhead; second, strengthening their capacity to adapt and maintain robust performance under domain shifts; and finally, understanding precisely why transformers—despite their success in NLP and Computer Vision—struggle to maintain their edge for long-horizon forecasting.

2.4 Foundation Models & Learning Representations

2.4.1 Connection with the thesis

Foundation models have achieved remarkable success across NLP and computer vision by learning powerful and generalizable representations from massive datasets. However, directly adapting these large-scale architectures to multivariate time series remains challenging due to computational resource constraints and data limitations. This thesis contributes to the emerging effort to leverage foundation models for time series through an innovative adaptation approach introduced in Chapter 5. Unlike classical foundation models—which often focus on extensive fine-tuning—we significantly compress latent representations, achieving near-original performance with drastically reduced complexity. This approach diverges from traditional foundation model methodologies reviewed here, as we explicitly address the bottleneck of computational feasibility in multivariate time series classification tasks, making powerful pre-trained representations accessible for broader practical use.

2.4.2 Introduction

Foundation models have had a transformative effect on numerous fields, most notably in computer vision (K. He et al., 2015; Dosovitskiy et al., 2021a) and NLP (Joshua Achiam et al., 2023; Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar,

et al., 2023a). By pre-training on large datasets, these models capture versatile representations that reduce the demand for extensive labeled data in downstream tasks (Bommasani et al., 2021). Building on this success, time series foundation models (TSFMs) have recently emerged, harnessing unstructured time series at scale to learn encoders transferable to forecasting, classification, imputation, and anomaly detection (Rasul, Anikait Ashok, et al., 2023; Das et al., 2024; G. Woo et al., 2024b; Yipeng Wang et al., 2024; T. Zhou, PeiSong Niu, et al., 2023; Goswami et al., 2024). Although these encoders are promising, a key challenge lies in efficiently modeling multivariate dependencies while minimizing running time and memory overhead.

2.4.3 Problem Setup

Let $\mathbf{X} = \{x_t\}_{t=1}^{T}$ be a time series with d channels, where each observation $x_t \in \mathbb{R}^d$. A *time series foundation model* (TSFM) is an encoder

$$F_\Psi : \mathbb{R}^{T \times d} \longrightarrow \mathbb{R}^q$$

parametrized by Ψ. It projects the time series data into a latent space of dimension q. During a *pre-training* phase, F_Ψ is trained on an unlabeled dataset $\mathcal{X}_0$ to learn *rich, transferable representations* that generalize well across diverse tasks. Once F_Ψ is pre-trained, we adapt it to a *downstream task* via a *fine-tuning* stage, typically using a labeled dataset $\{(\mathbf{X}, \mathbf{Y})\}$. Concretely, we introduce a *task-specific head*

$$h_\Phi : \mathbb{R}^q \longrightarrow \mathbb{R}^K$$

parametrized by Φ, where K is the dimension required by the task. For example, in time series classification with K classes, h_Φ outputs a vector of logits in $\mathbb{R}^K$. We thus compose the two components into a final model $h_\Phi \circ F_\Psi$.

Zero Shot Transfer. We *freeze* all parameters of the encoder F_Ψ (i.e., Ψ is not updated) and also freeze the task head h_Φ that was was part of the original pre-training. Zero-shot typically implies no further adaptation of the foundation model weights, thus reusing the pre-trained features directly.

Fine-Tuning Strategies. Depending on how we train or freeze the parameters Ψ and Φ, we distinguish three common scenarios:

- **Head-only Fine-Tuning:** We freeze the encoder parameters Ψ but learn a new head h_Φ for the downstream task by minimizing a suitable loss for the considered downstream task. This allows us to keep the pre-trained representations intact while adapting the final output layer to the specific classes or regression targets of the new task.

- **Full Fine-Tuning:** We update both Ψ and Φ on the downstream dataset. Concretely, we minimize a loss $\mathcal{L}(h_\Phi(F_\Psi(\mathbf{X})), \mathbf{Y})$ by backpropagating through *all* layers of the TSFM and the task head. This approach can yield higher accuracy but may require more data and careful hyperparameter tuning to avoid catastrophic forgetting or overfitting.

Figure 2.9 summarizes these three strategies in a flow diagram.

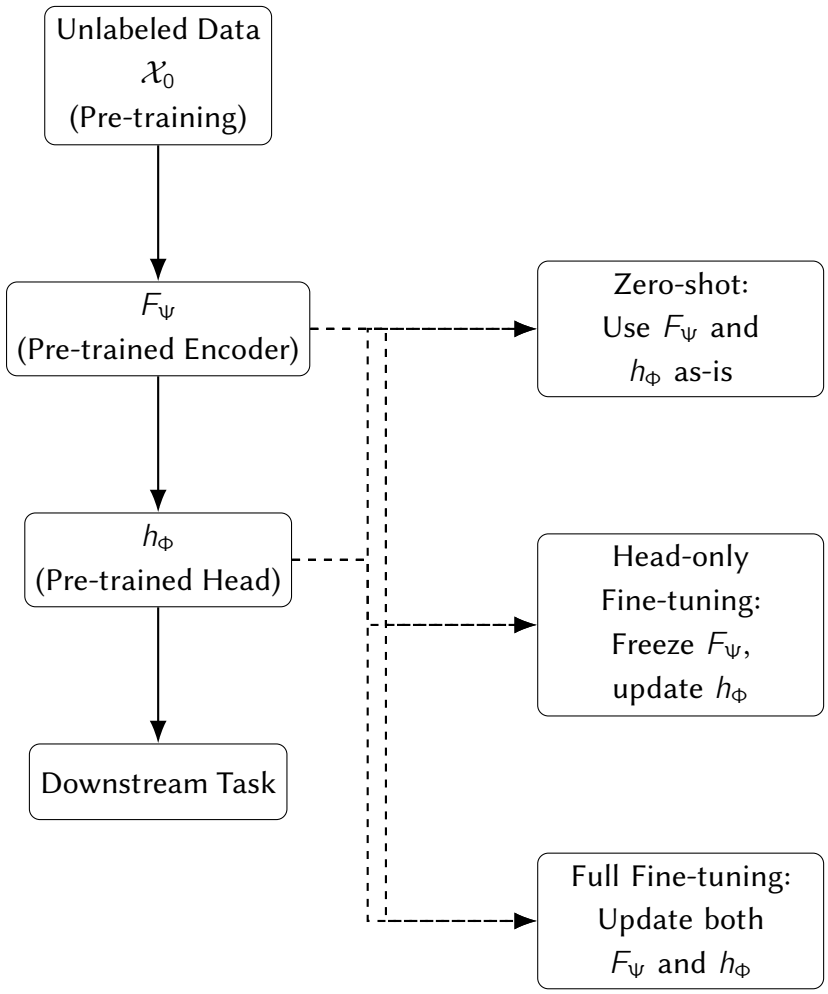


Figure 2.9: Flow diagram of different strategies for a time series foundation model. In the zero-shot approach, the pre-trained encoder F_Ψ and head h_Φ are used without modification. In head-only fine-tuning, only the head h_Φ is updated while F_Ψ remains frozen. In full fine-tuning, both F_Ψ and h_Φ are fine-tuned on the downstream task.

In summary, a TSFM provides a flexible pipeline for downstream tasks. We first learn a powerful encoder F_Ψ from unlabeled data, then compose it with a task head h_Φ for classification, forecasting, anomaly detection, or any other objective. Depending on resource constraints and data availability, we can choose a zero-shot approach (no additional training), head-only fine-tuning (lightweight adaptation), or full fine-tuning (maximum adaptation).

Pre-Training Strategies. Pre-training methods for time series representation learning aim to extract robust and generalizable features. Broadly, these methods can be categorized into two main approaches:

- **Contrastive Learning:** These methods train encoders to maximize similarity between augmented views of the same time series, while minimizing similarity across distinct series. By encouraging invariance to perturbations, contrastive methods produce representations resilient to noise and domain shifts. Prominent examples include TS2Vec (Yue et al., 2022) and Mantis (Feofanov, S. Wen, et al., 2024).
- **Masked and Reconstruction-Based Approaches:** These methods train encoders to reconstruct masked, missing, or future values from partial or corrupted inputs, implicitly capturing temporal structure and dynamics. Inspired by masked language modeling in NLP, models such as TNC (Tonekaboni et al., 2021), MOIRAI (G. Woo et al., 2024a), and MOMENT (Goswami et al., 2024) leverage this paradigm to improve contextual representation and forecasting performance.

Each method has advantages: contrastive learning is effective for classification, while masked modeling is suited for imputation and anomaly detection (T. Zhou, PeiSong Niu, et al., 2023).

2.4.4 Multivariate Time Series Challenges

In practice, time series data often consist of multiple channels, and the number of channels can vary greatly across different applications. This variability introduces several significant challenges for time series foundation models:

- **High Memory Consumption and Increased Running Time:** Many foundation models are computationally intensive. When processing all channels as a single input, the memory footprint becomes excessively high, especially with hundreds or thousands of channels. Alternatively, processing channels sequentially to reduce memory usage leads to a significant increase in running time.
- **Redundant or Correlated Channels:** In many multivariate time series, several channels are highly correlated or even redundant. Treating channels independently fails to exploit these interdependencies, leading to inefficient representations and a potential loss of valuable information. This redundancy can further exacerbate computational and memory issues.
- **Scalability Issues:** As the number of channels grows, the computational cost of processing each channel separately scales linearly (or worse). Without proper adaptation, applying a foundation model in a multivariate setting can result in prohibitive training times and memory requirements.

A promising solution to these difficulties is the use of a *channel-level adapter*, which compresses the original high-dimensional input into a more compact representation before passing it to the foundation model. By combining correlated channels, this adapter preserves the temporal structure, reduces the memory and runtime burden, and aligns the input with the model's computational constraints. Despite these advances, handling large-scale multivariate time series efficiently remains a challenge, particularly for memory-constrained environments.

2.4.5 Pre-Training Data and Benchmarks

A wide range of publicly available datasets has been curated to benchmark the performance of time series foundation models across diverse tasks, forecasting horizons, and domains. Below, we briefly summarize four major collections, along with their respective references.

Long-Horizon Forecasting Datasets (Informer). As introduced by (H. Zhou et al., 2021), the Informer suite of long-horizon forecasting datasets is composed of nine distinct time series that span various temporal resolutions and domains. This collection includes:

- **ETT (Electricity Transformer Temperature):** Available in hourly and minutely subsets, designed to test the ability of models to capture fine-grained temporal dynamics (H. Zhou et al., 2021).
- **Electricity:** Originating from Trindade 2015, this dataset tracks power consumption across multiple clients.
- **Traffic:** Released by the California Department of Transportation (California Department of Transportation, 2024), capturing vehicle flow rates on road segments.
- **Weather:** Provided by the Max Planck Institute for Biogeochemistry (Max Planck Institute for Biogeochemistry, 2024), containing meteorological measurements such as temperature, humidity, and wind speed.
- **ILI (Influenza-like Illness):** Published by the Centers for Disease Control and Prevention (Centers for Disease Control and Prevention, 2024), used to monitor and forecast flu trends.
- **Exchange-rate:** Introduced by (Guokun Lai et al., 2018b), focuses on multivariate exchange rate movements.

These datasets have been widely adopted to evaluate long-horizon forecasting performance (X. Wu et al., 2023; L. Nie et al., 2023; C. Challu et al., 2023). Interestingly, some recent studies report that foundation models based on transformers may offer superior performance on longer horizons, challenging earlier beliefs that simpler linear baselines dominate at scale.

Monash Time Series Forecasting Archive. Proposed by R. Godahewa et al. 2021, this archive consists of 58 publicly available short-horizon forecasting datasets, collectively containing over 100K time series. The archive encompasses diverse domains (e.g., finance, meteorology, health) and multiple temporal resolutions. Its broad coverage makes it an essential benchmark for evaluating methods intended for short-range predictions, especially those that emphasize operational decision-making (e.g., daily to weekly forecasts).

UCR/UEA Classification Archive. Time series classification remains a critical sub-task within time series analysis. The UCR/UEA archive (Dau et al., 2019) comprises 159 datasets spanning seven categories, such as image outlines, sensor readings, motion capture data, spectrographs, ECG recordings, electric devices, and simulated data. These datasets vary widely in both size and number of classes, enabling a comprehensive assessment of classification algorithms. They have served as a core benchmark for numerous classification studies (Ismail Fawaz, Forestier, et al., 2019).

TSB-UAD Anomaly Detection Benchmark. A more recent contribution, TSB-UAD (Paparrizos et al., 2022), provides 1980 univariate time series labeled with anomalies, drawn from 18 different anomaly detection datasets proposed over the past decade. Covering synthetic and real-world time series from sources such as human body signals, aerospace telemetry, environmental data, and web servers, TSB-UAD has quickly become a standard benchmark for anomaly detection research. Its extensive diversity in data characteristics (e.g., frequency, seasonality, trend) enables a rigorous evaluation of model robustness and generalization to novel anomaly types.

2.4.6 General Issues

Data Preprocessing. An essential prerequisite is rigorous data preprocessing: aligning series lengths, normalizing all signals, and ensuring consistent sets of variables across samples.

Task Mismatch. The concept of task mismatch arises from transfer learning. For instance, if a model is pre-trained on a classification objective, the features it learns may not be optimal for forecasting tasks. This mismatch can lead to suboptimal performance (S. J. Pan & Q. Yang, 2010).

In summary, while foundation models have the potential to excel on diverse downstream tasks—sometimes even outperforming models specifically designed for those tasks—they require careful fine-tuning. Addressing issues like data preprocessing and task mismatch is essential to achieving robust generalization.

Task	Dataset	Channels	Series Length	Data Size (Train, Val, Test)	Information (Frequency/Number of Classes)
Long horizon forecasting (Informer)	ETTm1, ETTm2	7	{96, 720}	(33953, 11425, 11425)	Electricity (15 mins)
	ETTh1, ETTh2	7		(8033, 2785, 2785)	Electricity (15 mins)
	Electricity	321		(17805, 2537, 5165)	Electricity (Hourly)
	Traffic	862		(11673, 1661, 3413)	Transportation (Hourly)
	Weather	21		(36280, 5175, 10444)	Weather (10 mins)
	Exchange	8		(4704, 665, 1422)	Exchange rate (Daily)
	ILI	7	{24, 60}	(69, 2, 98)	Illness (Weekly)
Short horizon forecasting (Monash)	M4-Yearly	1	6	(16099, 2301, 4600)	-
	M4-Quarterly		8	(16800, 2400, 4800)	-
	M4-Monthly		18	(33600, 4800, 9600)	-
	M3-Yearly		6	(451, 65, 129)	-
	M3-Quarterly		8	(529, 76, 151)	-
	M3-Monthly		18	(999, 144, 285)	-
Imputation (Informer)	ETTm1, ETTm2	7	512	(33953, 11425, 11425)	Electricity (15 mins)
	ETTh1, ETTh2	7		(8033, 2785, 2785)	Electricity (15 mins)
	Electricity	321		(17805, 2537, 5165)	Electricity (Hourly)
	Weather	21		(36280, 5175, 10444)	Weather (10 mins)
Classification (UCR)	UWaveGestureLibraryX	1	315	(640, 256, 3582)	Motion Gesture (8 classes)
	ECG5000		140	(357, 143, 4500)	ECG Record (5 classes)
	OSULeaf		427	(142, 58, 242)	Leaf Outlines (6 classes)
	MedicalImages		99	(272, 109, 760)	Pixel Intensity (10 classes)
	Ham		431	(77, 32, 105)	Food spectrographs (2 classes)
Anomaly detection (TSB-UAD)	1sddb40	1	-	(24489, 9489, 3969)	Beats
	BIDMC1		-	(1274, 204, 7988)	PVC
	CIMIS44AirTemperature3		-	(2346, 632, 3672)	Weather Data
	CIMIS44AirTemperature5		-	(2346, 632, 3672)	Weather Data
	ECG2		-	(10203, 3775, 14488)	ECG2 Lead

Figure 2.10: Adapted from the MOMENT paper (Goswami et al., 2024). A brief description of the datasets that collectively form the *Time Series Pile*. Due to space constraints, the authors only provide metadata for the subsets of the M3 and M4 datasets used in their experiments, along with five classification and anomaly detection datasets. Detailed characteristics for all short-horizon forecasting, classification, and anomaly detection datasets in the Time Series Pile can be found in the official repository, as well as in the Monash archive (R. Godahewa et al., 2021), the UCR/UEA classification archive (Dau et al., 2019), and the TSB-UAD anomaly benchmark (Paparrizos et al., 2022).

2.4.7 Conclusions

TSFMs are an emerging research area that leverages large-scale pre-training to capture rich, general-purpose representations. Models such as MOIRAI (G. Woo et al., 2024a) demonstrate that, when fine-tuned properly, these models can outperform task-specific architectures. However, challenges require the use of strategies to prevent overfitting and loss of pre-trained knowledge. Future work should explore more robust fine-tuning strategies and cross-task generalization methods.

Chapter 3

SAMformer: Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention

Summary. Transformer-based architectures achieved breakthrough performance in natural language processing and computer vision, yet they remain inferior to simpler linear baselines in multivariate long-term forecasting. To better understand this phenomenon, we start by studying a toy linear forecasting problem for which we show that transformers are incapable of converging to their true solution despite their high expressive power. We further identify the attention of transformers as being responsible for this low generalization capacity. Building upon this insight, we propose a shallow lightweight transformer model that successfully escapes bad local minima when optimized with sharpness-aware optimization. We empirically demonstrate that this result extends to all commonly used real-world multivariate time series datasets. In particular, SAMformer surpasses current state-of-the-art methods and is on par with the biggest foundation model MOIRAI while having significantly fewer parameters. The code is available at https://github.com/romilbert/samformer.

3.1 Introduction

Multivariate time series forecasting is a classical learning problem that consists of analyzing time series to predict future trends based on historical information. In particular, long-term forecasting is notoriously challenging due to feature correlations and long-term temporal dependencies in time series. This learning problem is prevalent in those real-world applications where observations are gathered sequentially, such as medical data (Čepulionis & Lukoševičiūtė, 2016), electricity consumption (UCI, 2015), temperatures (Max Planck

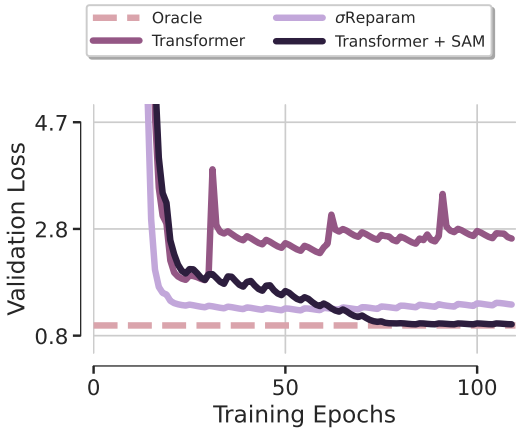


Figure 3.1: Illustration of our approach on synthetic data. Oracle is the optimal solution, Transformer is a base transformer, σReparam is a Transformer with weight rescaling (Zhai et al., 2023) and Transformer + SAM is Transformer trained with sharpness-aware minimization. Transformer overfits, σReparam improves slightly but fails to reach Oracle while Transformer+SAM generalizes perfectly. This motivates SAMformer, a shallow transformer combining SAM and best practices in time series forecasting.

Institute, 2021), or stock prices (Sonkavde et al., 2023). A plethora of methods have been developed for this task, from classical mathematical tools (Sorjamaa et al., 2007; R. Chen & M. Tao, 2021) and statistical approaches like ARIMA (George Edward Pelham Box & G. Jenkins, 1990; G. E. P. Box et al., 1974) to more recent deep learning ones (Casolaro et al., 2023), including recurrent and convolutional neural networks (Rangapuram et al., 2018; Salinas et al., 2020b; Fan et al., 2019; Guokun Lai et al., 2018c; Sen et al., 2019).

Recently, the transformer architecture (Vaswani et al., 2017) became ubiquitous in natural language processing (NLP) (Devlin et al., 2018; Radford et al., 2018; Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al., 2023b; OpenAI, 2023) and computer vision (Dosovitskiy et al., 2021b; Caron et al., 2021; Touvron, Cord, et al., 2021), achieving breakthrough performance in both domains. Transformers are known to be particularly efficient in dealing with sequential data, a property that naturally calls for their application on time series. Unsurprisingly, many works attempted to propose time series-specific transformer architectures to benefit from their capacity to capture temporal interactions (H. Zhou et al., 2021; H. Wu et al., 2021; T. Zhou, Ma, et al., 2022; Yuqi Nie et al., 2023). However, the current state-of-the-art in multivariate time series forecasting is achieved with a simpler MLP-based model (S.-A. Chen et al., 2023), which significantly outperforms transformer-based methods. Moreover, Zeng, M. Chen, et al. 2023 have recently found that linear networks can be on par or better than transformers for the forecasting task, questioning their practical utility. This curious finding serves as a starting point for our work.

Limitation of current approaches. Recent works applying transformers to time series data have mainly focused on either (i) efficient implementations reducing the quadratic

cost of attention (Shiyang Li et al., 2019; S. Liu et al., 2022; Cirstea et al., 2022; Kitaev et al., 2020; H. Zhou et al., 2021; H. Wu et al., 2021) or (ii) decomposing time series to better capture the underlying patterns in them (H. Wu et al., 2021; T. Zhou, Ma, et al., 2022). Surprisingly, none of these works have specifically addressed a well-known issue of transformers related to their training instability, particularly present in the absence of large-scale data (L. Liu et al., 2020; Dosovitskiy et al., 2021b).

Trainability of transformers. In computer vision and NLP, it has been found that attention matrices can suffer from entropy or rank collapse (Dong et al., 2021). Then, several approaches have been proposed to overcome these issues X. Chen et al., 2022; Zhai et al., 2023. However, in the case of time series forecasting, open questions remain about how transformer architectures can be trained effectively without a tendency to overfit. We aim to show that by eliminating training instability, transformers can excel in multivariate long-term forecasting, contrary to previous beliefs of their limitations.

Summary of our contributions. Our proposal puts forward the following contributions:

1. We show that even when the transformer architecture is tailored to solve a simple toy linear forecasting problem, it still generalizes poorly and converges to sharp local minima. We further identify that attention is mainly responsible for this phenomenon;
2. We propose a shallow transformer model, termed SAMformer, that incorporates the best practices proposed in the research community including reversible instance normalization (RevIN, T. Kim et al. 2021) and channel-wise attention H. Zhang et al., 2022; Zamir et al., 2022 recently introduced in computer vision community. We show that optimizing such a simple transformer with sharpness-aware minimization (SAM) allows convergence to local minima with better generalization;
3. We empirically demonstrate the superiority of our approach on common multivariate long-term forecasting datasets. SAMformer surpasses current state-of-the-art methods and is on par with the biggest foundation model MOIRAI while having significantly fewer parameters.

3.2 Proposed Approach

Notations. We represent scalar values with regular letters (e.g., parameter λ), vectors with bold lowercase letters (e.g., vector $\mathbf{x}$), and matrices with bold capital letters (e.g., matrix $\mathbf{M}$). We denote by $\mathbf{M}^\top$ the transpose of $\mathbf{M}$ and likewise for vectors. The rank of a matrix $\mathbf{M}$ is denoted by $\mathrm{rank}(\mathbf{M})$, and its Frobenius norm by $\|\mathbf{M}\|_\mathrm{F}$. We let $\tilde{n} = \min\{n, m\}$, and denote by $\|\mathbf{M}\|_* = \sum_{i=1}^{\tilde{n}} \sigma_i(\mathbf{M})$ the nuclear norm of $\mathbf{M}$ with $\sigma_i(\mathbf{M})$ being its singular

values, and by $\|\mathbf{M}\|_2 = \sigma_{\max}(\mathbf{M})$ its spectral norm. The identity matrix of size $n \times n$ is denoted by $\mathbf{I}_n$. The notation $\mathbf{M} \succcurlyeq \mathbf{0}$ indicates that $\mathbf{M}$ is positive semi-definite.

3.2.1 Problem Setup

We consider the multivariate long-term forecasting framework: given a D-dimensional time series of length L (*look-back window*), arranged in a matrix $\mathbf{X} \in \mathbb{R}^{D \times L}$ to facilitate channel-wise attention, our objective is to predict its next H values (*prediction horizon*), denoted by $\mathbf{Y} \in \mathbb{R}^{D \times H}$. We assume that we have access to a training set that consists of N observations $(\mathcal{X}, \mathcal{Y}) = (\{\mathbf{X}^{(i)}\}_{i=0}^{N}, \{\mathbf{Y}^{(i)}\}_{i=0}^{N})$, and denote by $\mathbf{X}_d^{(i)} \in \mathbb{R}^{1 \times L}$ (respectively $\mathbf{Y}_d^{(i)} \in \mathbb{R}^{1 \times H}$) the d-th feature of the i-th input (respectively target) time series. We aim to train a predictor $f_{\boldsymbol{\omega}} : \mathbb{R}^{D \times L} \to \mathbb{R}^{D \times H}$ parameterized by $\boldsymbol{\omega}$ that minimizes the mean squared error (MSE) on the training set:

$$\mathcal{L}_{\text{train}}(\boldsymbol{\omega}) = \frac{1}{ND} \sum_{i=0}^{N} \|\mathbf{Y}^{(i)} - f_{\boldsymbol{\omega}}(\mathbf{X}^{(i)})\|_{\text{F}}^2 . \tag{3.1}$$

3.2.2 Motivational Example

Recently, Zeng, M. Chen, et al. 2023 showed that transformers perform on par with, or are worse than, simple linear neural networks trained to directly project the input to the output. We use this observation as a starting point by considering the following generative model for our toy regression problem mimicking a time series forecasting setup considered later:

$$\mathbf{Y} = \mathbf{X}\mathbf{W}_{\text{toy}} + \boldsymbol{\varepsilon}. \tag{3.2}$$

We let $L = 512$, $H = 96$, $D = 7$ and $\mathbf{W}_{\text{toy}} \in \mathbb{R}^{L \times H}$, $\boldsymbol{\varepsilon} \in \mathbb{R}^{D \times H}$ having random normal entries and generate 15000 input-target pairs $(\mathbf{X}, \mathbf{Y})$ (10000 for train and 5000 for validation), with $\mathbf{X} \in \mathbb{R}^{D \times L}$ having random normal entries.

Given this generative model, we would like to develop a transformer architecture that can efficiently solve the problem in Eq. (3.2) without unnecessary complexity. To achieve this, we propose to simplify the usual transformer encoder by applying attention to $\mathbf{X}$ and incorporating a residual connection that adds $\mathbf{X}$ to the attention's output. Instead of adding a feedforward block on top of this residual connection, we directly employ a linear layer for output prediction. Formally, our model is defined as follows:

$$f(\mathbf{X}) = [\mathbf{X} + \mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}_V\mathbf{W}_O]\mathbf{W}, \tag{3.3}$$

with $\mathbf{W} \in \mathbb{R}^{L \times H}$, $\mathbf{W}_V \in \mathbb{R}^{L \times d_{\text{m}}}$, $\mathbf{W}_O \in \mathbb{R}^{d_{\text{m}} \times L}$ and $\mathbf{A}(\mathbf{X})$ being the *attention matrix* of an input sequence $\mathbf{X} \in \mathbb{R}^{D \times L}$ defined as

$$\mathbf{A}(\mathbf{X}) = \text{softmax}\left(\frac{\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top}{\sqrt{d_{\text{m}}}}\right) \in \mathbb{R}^{D \times D} \tag{3.4}$$

where the softmax is row-wise, $\mathbf{W}_Q \in \mathbb{R}^{L \times d_m}$, $\mathbf{W}_K \in \mathbb{R}^{L \times d_m}$, and d_m is the dimension of the model. The softmax makes $\mathbf{A}(\mathbf{X})$ right stochastic, with each row describing a probability distribution. To ease the notations, in contexts where it is unambiguous, we refer to the attention matrix simply as $\mathbf{A}$, omitting $\mathbf{X}$.

We term this architecture Transformer and briefly comment on it. First, the attention matrix is applied channel-wise, which simplifies the problem and reduces the risk of over-parametrization, as the matrix $\mathbf{W}$ has the same shape as in Eq. (3.2) and the attention matrix becomes much smaller due to $L > D$. In addition, channel-wise attention is more relevant than temporal attention in this scenario, as data generation follows an i.i.d. process according to Eq. (3.2). We formally establish the identifiability of $\mathbf{W}_{\text{toy}}$ by our model below. The detailed proof, including all supporting lemmas, is deferred to Appendix A.2.2.

Proposition 3.2.1 (Existence of optimal solutions). *Assume $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ and $\mathbf{W}_O$ are fixed and let $\mathbf{P} = \mathbf{X} + \mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}_V\mathbf{W}_O \in \mathbb{R}^{D \times L}$. Then, there exists a matrix $\mathbf{W} \in \mathbb{R}^{L \times H}$ such that $\mathbf{P}\mathbf{W} = \mathbf{X}\mathbf{W}_{toy}$ if, and only if, $\text{rank}([\mathbf{P} \quad \mathbf{X}\mathbf{W}_{\text{toy}}]) = \text{rank}(\mathbf{P})$ where $[\mathbf{P} \quad \mathbf{X}\mathbf{W}_{\text{toy}}] \in \mathbb{R}^{D \times (L+H)}$ is a block matrix.*

We now proceed to the proof of Proposition 3.2.1.

Proof. Applying Lemma A.2.2 with $\mathbf{S} = \mathbf{P}$, $\mathbf{B} = \mathbf{0}$, $\mathbf{C} = \mathbf{X}\mathbf{W}_{\text{toy}}$ and $\mathbf{W}$ in the role of $\mathbf{Y}$ ensures that there exists $\mathbf{W} \in \mathbb{R}^{L \times H}$ such that $\mathbf{P}\mathbf{W} = \mathbf{X}\mathbf{W}_{\text{toy}}$ if and only if $\text{rank}([\mathbf{P} \quad \mathbf{X}\mathbf{W}_{\text{toy}}]) = \text{rank}(\mathbf{P})$, which concludes the proof. □

The assumption made above is verified if P is full rank and $D < H$, which is the case in this toy experiment. Consequently, the optimization problem of fitting a transformer on data generated with Eq. (3.2) theoretically admits infinitely many optimal classifiers $\mathbf{W}$.

We would now like to identify the role of attention in solving the problem from Eq. (3.3). To this end, we consider a model, termed Random Transformer, where only $\mathbf{W}$ is optimized, while self-attention weights $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$ are fixed during training and initialized following Glorot & Bengio 2010. This effectively makes the considered transformer act like a linear model. Finally, we compare the local minima obtained by these two models after their optimization using Adam with the Oracle model that corresponds to the least squares solution of Eq. (3.2).

We present the validation loss for both models in Figure 3.2. A first surprising finding is that both transformers fail to recover $\mathbf{W}_{\text{toy}}$, highlighting that optimizing even such a simple architecture with a favorable design exhibits a strong lack of generalization. When fixing the self-attention matrices, the problem is alleviated to some extent, although Random Transformer remains suboptimal. This observation remains consistent across various optimizers (see Figure 3.18) and values of learning rate, suggesting that this phenomenon is not attributable to suboptimal optimizer hyperparameters or the specific choice of the optimizer. As there is only a 2% increase in the number of parameters between the Random Transformer and the Transformer, it is not due to overfitting either. Hence, we deduce

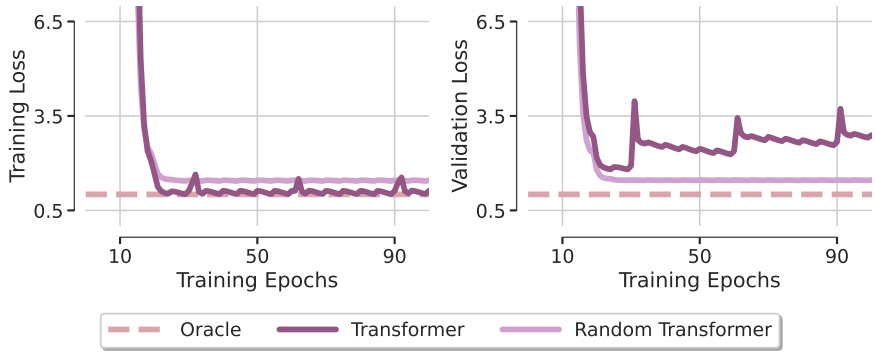


Figure 3.2: Poor generalization. Despite its simplicity, Transformer suffers from severe overfitting. Fixing the attention weights in Random Transformer improves the generalization, hinting at the role of attention in preventing convergence to optimal local minima.

from Figure 3.1 that the poor generalization capabilities of Transformer are mostly due to the trainability issues of the attention module.

3.2.3 Transformer's Loss Landscape

Intuition. In the previous section, we concluded that the attention was at fault for the poor generalization of Transformer observed above. To develop our intuition behind this phenomenon, we plot in Figure 3.3 the attention matrices at different epochs of training. We can see that the attention matrix is close to the identity matrix right after the very first epoch and barely changes afterward, especially with the softmax amplifying the differences in the matrix values. It shows the emergence of *attention's entropy collapse* with a full-rank attention matrix, which was identified in Zhai et al. 2023 as one of the reasons behind the hardness of training transformers. This work also establishes a relationship between entropy collapse and the sharpness of the transformers' loss landscape, which we confirm in Figure 3.4 (a similar behavior is obtained on real data in Figure 3.6. The Transformer converges to a sharper minimum than the Random Transformer while having a significantly lower entropy (the attention being fixed at initialization for the latter, its entropy remains constant along training). These pathological patterns suggest that the Transformer fails because of the entropy collapse and the sharpness of its training loss. In the next paragraph, we investigate the existing solutions in the literature to alleviate those issues.

Existing solutions. Recent studies have demonstrated that the loss landscape of transformers is sharper compared to other residual architectures (X. Chen et al., 2022; Zhai et al., 2023). This may explain training instability and subpar performance of transformers, especially when trained on small-scale datasets. The sharpness of transformers was observed and quantified differently: while X. Chen et al. 2022 computes $\lambda_{\max}$, the largest eigenvalue of the loss function's Hessian, Zhai et al. 2023 gauges the entropy of the attention matrix to demonstrate its collapse with high sharpness. Both these metrics are evaluated, and their

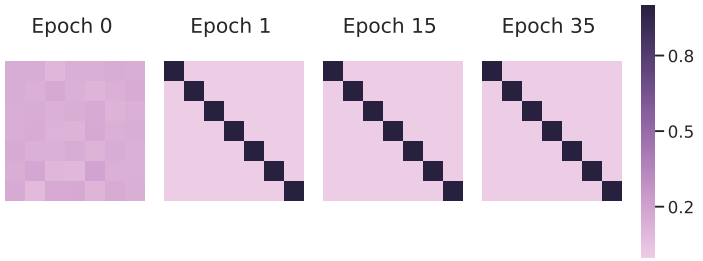


Figure 3.3: Transformer's loss landscape analysis for linear regression. The attention matrices of Transformer quickly become fixed to the identity from the very first epoch, indicating a lack of dynamic adaptation during training.

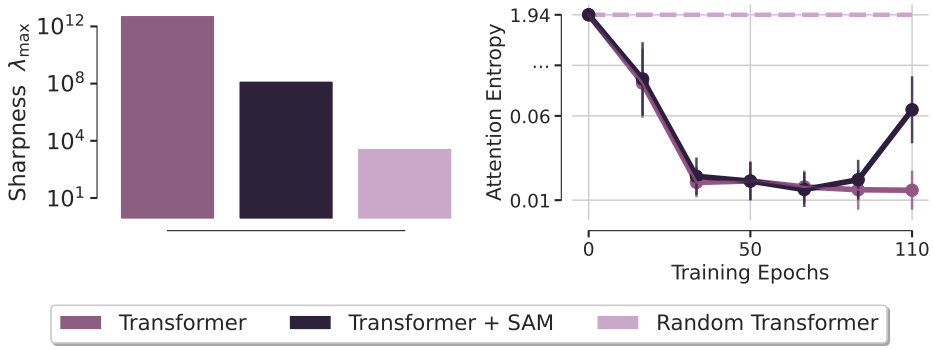


Figure 3.4: Analysis of the loss landscape at the end of training. (Left) Transformer converges to a much sharper minimum than Transformer+SAM, as evidenced by a significantly larger λ_{max} (approximately 10^4 times larger), whereas the Random Transformer exhibits a smoother loss landscape. (Right) Transformer experiences entropy collapse during training, further confirming the high sharpness of its loss landscape.

results are illustrated in Figure 3.4. This visualization confirms our hypothesis, revealing both detrimental phenomena at once. On the one hand, the sharpness of the transformer with fixed attention is orders of magnitude lower than the sharpness of the transformer that converges to the identity attention matrix. On the other hand, the entropy of the transformer's attention matrix is dropping sharply along the epochs when compared to the initialization.

To identify an appropriate solution allowing a better generalization performance and training stability, we explore both remedies proposed by X. Chen et al. 2022 and Zhai et al. 2023. The first approach involves utilizing the recently proposed sharpness-aware minimization framework (Foret et al., 2021) which replaces the training objective $\mathcal{L}_{\text{train}}$ of Eq. (3.1) by

$$\mathcal{L}^{\text{SAM}}_{\text{train}}(\boldsymbol{\omega}) = \max_{\|\boldsymbol{\epsilon}\|<\rho} \mathcal{L}_{\text{train}}(\boldsymbol{\omega} + \boldsymbol{\epsilon}) \,,$$

where $\rho > 0$ is an hyper-parameter (see Remark A.1.1 of Appendix A.1), and $\boldsymbol{\omega}$ are the parameters of the model. More details on SAM can be found in Appendix A.1.2. The second approach involves reparameterizing all weight matrices with spectral normalization and an

additional learned scalar, a technique termed σReparam by Zhai et al. 2023. More formally, we replace each weight matrix $\mathbf{W}$ as follows

$$\widehat{\mathbf{W}} = \frac{\gamma}{\|\mathbf{W}\|_2}\mathbf{W}, \tag{3.5}$$

where $\gamma \in \mathbb{R}$ is a learnable parameter initialized at 1.

The results depicted in Figure 3.1 highlight our transformer's successful convergence to the desired solution. Surprisingly, this is only achieved with SAM, as σReparam doesn't manage to approach the optimal performance despite maximizing the entropy of the attention matrix. In addition, one can observe in Figure 3.4 that the sharpness with SAM is several orders of magnitude lower than the Transformer while the entropy of the attention obtained with SAM remains close to that of a base Transformer with a slight increase in the later stages of the training. It suggests that entropy collapse as introduced in Zhai et al. 2023 is benign in this scenario.

To better understand the failure of σReparam, it can be useful to recall how Eq. (3.5) was derived. Zhai et al. 2023 departed from a tight lower bound on the attention entropy and showed that it increases exponentially fast when $\|\mathbf{W}_Q\mathbf{W}_K^\top\|_2$ is minimized (Zhai et al., 2023, see Theorem 3.1). Eq. (3.5) was proposed as a simple way to minimize this quantity. In the case of channel-wise attention, however, it can be shown that this has a detrimental effect on the rank of the attention matrix, which would consequently exclude certain features from being considered by the attention mechanism. We formalize this intuition in the following Proposition 3.2.2, where we consider the nuclear norm, a sum of the singular values, as a smooth proxy of the algebraic rank, which is a common practice (Daneshmand et al., 2020; Dong et al., 2021). The detailed proof, including all supporting lemmas, is deferred to Appendix A.2.3.

Proposition 3.2.2 (Upper bound on the nuclear norm). *Let $\mathbf{X} \in \mathbb{R}^{D\times L}$ be an input sequence. Assuming $\mathbf{W}_Q\mathbf{W}_K^\top = \mathbf{W}_K\mathbf{W}_Q^\top \succcurlyeq \mathbf{0}$, we have*

$$\|\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\|_* \leq \|\mathbf{W}_Q\mathbf{W}_K^\top\|_2\|\mathbf{X}\|_{\mathrm{F}}^2.$$

Proof. Let $\mathbf{M} := \mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top$. Since $\mathbf{W}_Q\mathbf{W}_K^\top$ is symmetric and positive semi-definite (PSD), $\mathbf{M}$ is also PSD by Lemma A.2.6. For PSD matrices, the nuclear norm equals the trace (Lemma A.2.5), hence:

$$\|\mathbf{M}\|_* = \mathrm{Tr}(\mathbf{M}) = \mathrm{Tr}\big(\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\big) = \mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\big), \tag{3.6}$$

where we used the cyclic property of the trace.

We then apply the trace inequality from Lemma A.2.3:

$$\mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\big) \;\leq\; \lambda_{\max}\big(\mathbf{W}_Q\mathbf{W}_K^\top\big)\;\mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\big). \tag{3.7}$$

Since $\mathbf{W}_Q\mathbf{W}_K^\top$ is PSD, its largest eigenvalue equals its spectral norm by Lemma A.2.5:

$$\lambda_{\max}\left(\mathbf{W}_Q\mathbf{W}_K^\top\right) = \|\mathbf{W}_Q\mathbf{W}_K^\top\|_2. \tag{3.8}$$

Additionally, by definition of the Frobenius norm, we have:

$$\mathrm{Tr}\left(\mathbf{X}^\top\mathbf{X}\right) = \|\mathbf{X}\|_{\mathrm{F}}^2. \tag{3.9}$$

Combining these results yields the desired inequality:

$$\|\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\|_* \;\leq\; \|\mathbf{W}_Q\mathbf{W}_K^\top\|_2\,\|\mathbf{X}\|_{\mathrm{F}}^2, \tag{3.10}$$

which concludes the proof. □

Note that the assumption made above holds when $\mathbf{W}_Q = \mathbf{W}_K$ and has been previously studied by H. Kim et al. 2021. The theorem confirms that employing σReparam to decrease $\|\mathbf{W}_Q\mathbf{W}_K^\top\|_2$ reduces the nuclear norm of the numerator of attention matrix defined by Eq. (3.4). While the direct link between matrix rank and this nuclear norm does not always hold, nuclear norm regularization is commonly used to encourage a low-rank structure in compressed sensing (Recht et al., 2010; Recht, 2011; Candès & Recht, 2012).

Although Proposition 3.2.2 cannot be directly applied to the attention matrix $\mathbf{A}(\mathbf{X})$, we point out that in the extreme case when σReparam leads to the attention scores $\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top$ to be rank-1 with identical rows as studied in (Anagnostidis et al., 2022), that the attention matrix stays rank-1 after application of the row-wise softmax. Thus, σReparam may induce a collapse of the attention rank that we empirically observe in terms of nuclear norm in Figure 3.11. With these findings, we present a new simple transformer model with high performance and training stability for multivariate time series forecasting.

3.2.4 SAMformer: Putting It All Together

The proposed SAMformer is based on Eq. (3.3) with two important modifications. First, we equip it with Reversible Instance Normalization (RevIN, T. Kim et al. 2021) applied to $\mathbf{X}$ as this technique was shown to be efficient in handling the shift between the training and testing data in time series. Second, as suggested by our explorations above, we optimize the model with SAM to make it converge to flatter local minima. Overall, this gives the shallow transformer model with one encoder in Figure 3.5.

We highlight that SAMformer keeps the channel-wise attention represented by a matrix $D \times D$ as in Eq. (3.3), contrary to spatial (or temporal) attention given by $L \times L$ matrix used in other models. This brings two important benefits: (i) it ensures feature permutation invariance, eliminating the need for positional encoding, commonly preceding the attention layer; (ii) it leads to a reduced time and memory complexity as $D \leq L$ in most of the real-world datasets. Our channel-wise attention examines the average impact of

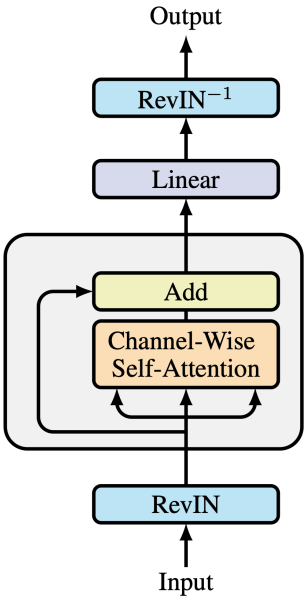


Figure 3.5: SAMformer Architecture

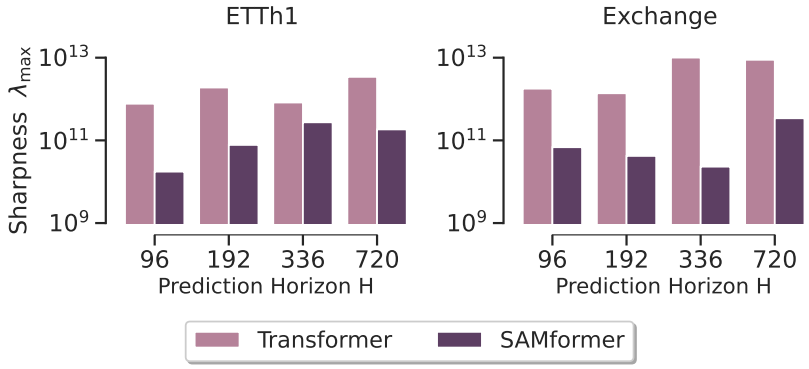
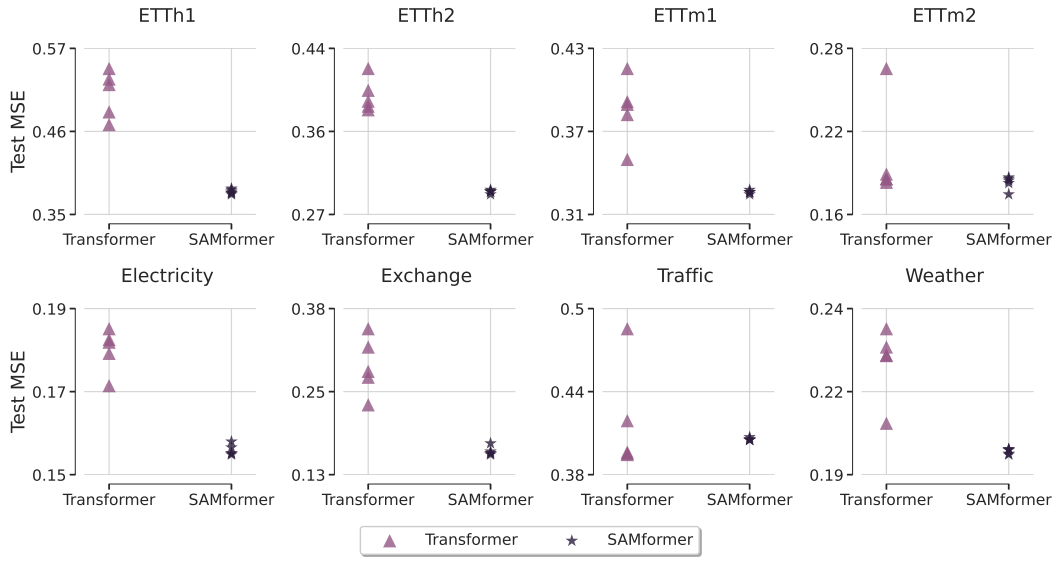


Figure 3.6: Sharpness of SAMformer and Transformer. This figure demonstrates that SAMformer exhibits a smoother loss landscape compared to Transformer.

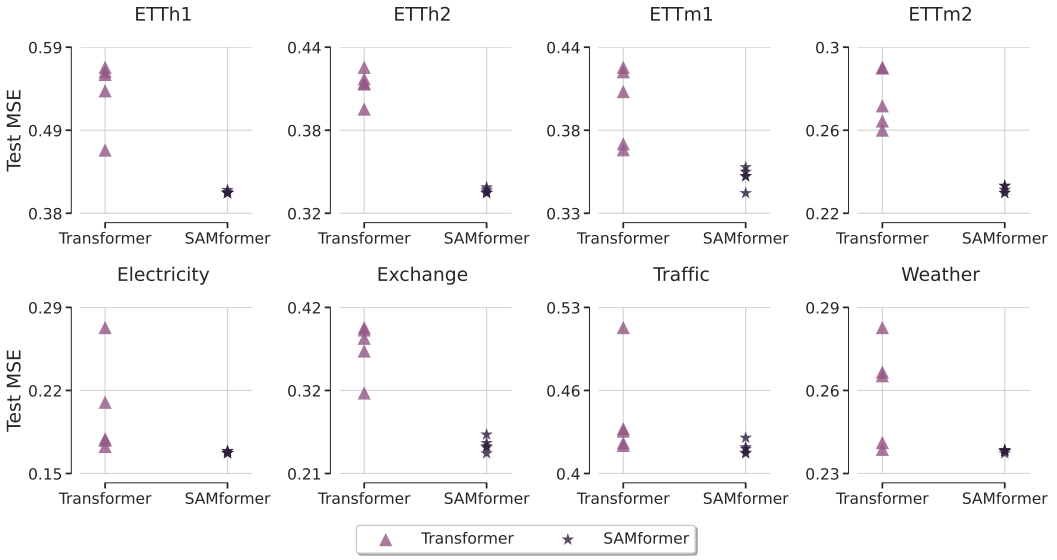
each feature on the others throughout all timesteps. An ablation study, detailed in Section 3.3.4, validates the effectiveness of this implementation. We are now ready to evaluate SAMformer on common multivariate time series forecasting benchmarks, demonstrating its superior performance.

3.3 Experiments

In this section, we empirically demonstrate the quantitative and qualitative superiority of SAMformer in multivariate long-term time series forecasting on common benchmarks. We



(a) Prediction horizon $H = 96$.

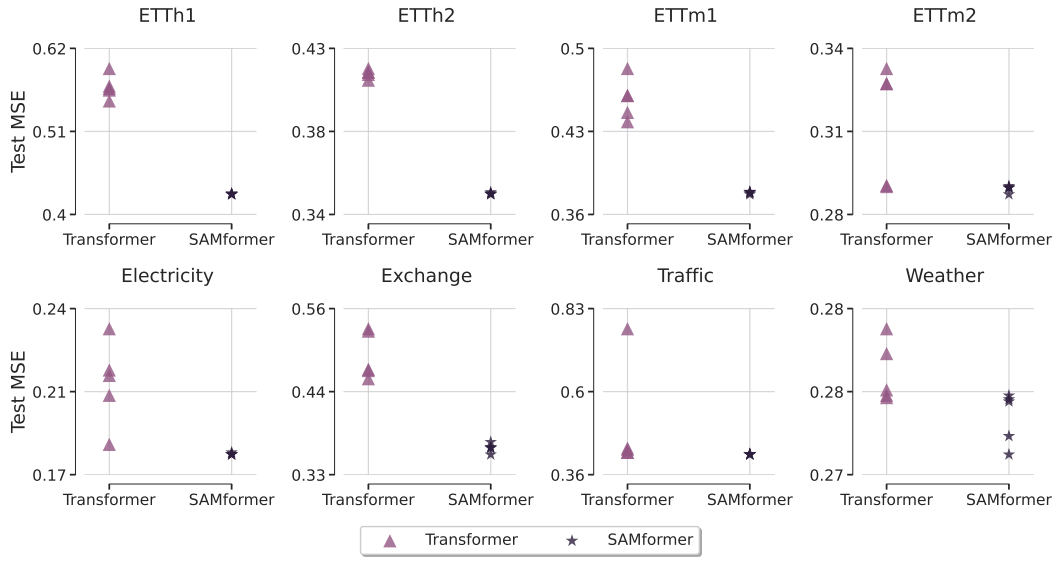


(b) Prediction horizon $H = 192$.

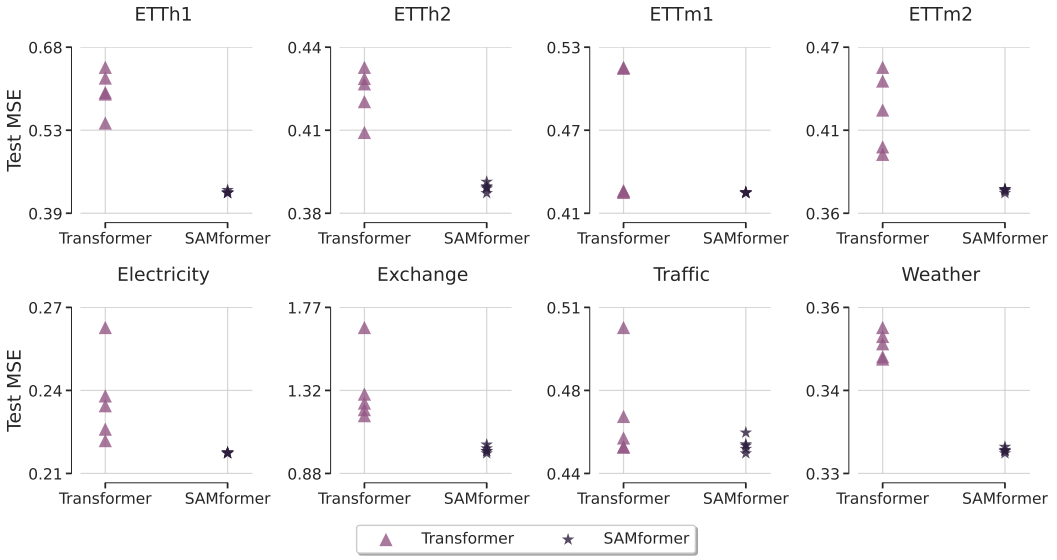
Figure 3.7: Test Mean Squared error on all datasets for a prediction horizon $H \in \{96, 192\}$ across five different seed values for Transformer and SAMformer. This plot reveals a significant variance for the Transformer, as opposed to the minimal variance of SAMformer, showing the high impact of weight initialization on Transformer and the high resilience of SAMformer.

show that SAMformer surpasses the current multivariate state-of-the-art TSMixer (S.-A. Chen et al., 2023) by 14.33% while having ~ 4 times fewer parameters.

Architecture. We follow S.-A. Chen et al. 2023; Yuqi Nie et al. 2023, and to ensure a fair comparison of baselines, we apply the reversible instance normalization (RevIN) of T. Kim



(a) Prediction horizon $H = 336$.



(b) Prediction horizon $H = 720$.

Figure 3.8: Test Mean Squared error on all datasets for a prediction horizon $H \in \{336, 720\}$ across five different seed values for Transformer and SAMformer. This plot reveals a significant variance for the Transformer, as opposed to the minimal variance of SAMformer, showing the high impact of weight initialization on Transformer and the high resilience of SAMformer.

et al. 2021 (see Appendix A.1.1 for more details). The network used in SAMformer and Transformer is a simplified one-layer transformer with one head of attention and without feed-forward. Its neural network function follows Eq. (3.3), while RevIN normalization and denormalization are applied respectively before and after the neural network function; see Figure 3.5. We display the inference step of SAMformer in detail in Algorithm 1. For the sake of clarity, we describe the application of the neural network function sequentially

on each element of the batches, but in practice, the operations are parallelized and performed batch per batch. For SAMformer and Transformer, the dimension of the model is $d_{\text{m}} = 16$ and remains the same in all our experiments. For TSMixer, we used the official implementation that can be found here.

Algorithm 1: Architecture of the network used in SAMformer and Transformer

Parameters: Batch size bs, input length L, prediction horizon H, dimension of the model d_{m}.

Network trainable parameters: $\mathbf{W}_Q \in \mathbb{R}^{L \times d_{\text{m}}}$, $\mathbf{W}_K \in \mathbb{R}^{L \times d_{\text{m}}}$, $\mathbf{W}_V \in \mathbb{R}^{L \times d_{\text{m}}}$, $\mathbf{W}_O \in \mathbb{R}^{d_{\text{m}} \times L}$, $\mathbf{W} \in \mathbb{R}^{L \times H}$.

RevIN trainable parameters: $\boldsymbol{\beta}, \boldsymbol{\gamma}$.

Input: Batch of bs input sequences $\mathbf{X} \in \mathbb{R}^{D \times L}$ arranged in a tensor $\mathbf{B}_{\text{in}}$ of dimension $bs \times L \times D$.

RevIN normalization: $\mathbf{X} \leftarrow \tilde{\mathbf{X}}$ following Eq. (A.2). The output is a tensor $\tilde{\mathbf{B}}_{\text{in}}$ of dimension $bs \times L \times D$.

Transposition of the batch: $\tilde{\mathbf{B}}_{\text{in}}$ is reshaped in dimension $bs \times D \times L$.

Applying the neural network of Eq. (3.3):

for *each* $\tilde{\mathbf{X}} \in \tilde{\mathbf{B}}_{\text{in}}$ **do**

> **1. Attention layer**
>
> Rescale the input with the attention matrix (Eq. (3.4)).
>
> The output $\mathbf{A}(\tilde{\mathbf{X}})\tilde{\mathbf{X}}\mathbf{W}_V\mathbf{W}_O$ is of dimension $D \times L$
>
> **2. Skip connection**
>
> Sum the input $\tilde{\mathbf{X}}$ and the output of the attention layer.
>
> The output $\tilde{\mathbf{X}} + \mathbf{A}(\tilde{\mathbf{X}})\tilde{\mathbf{X}}\mathbf{W}_V\mathbf{W}_O$ is of dimension $D \times L$.
>
> **3. Linear layer**
>
> Apply a linear layer on the output of the skip connection.
>
> The output $\tilde{\mathbf{Y}} = \left[\tilde{\mathbf{X}} + \mathbf{A}(\tilde{\mathbf{X}})\tilde{\mathbf{X}}\mathbf{W}_V\mathbf{W}_O\right]\mathbf{W}$ is of dimension $D \times H$.
>
> Unnormalized predictions are arranged in a tensor $\tilde{\mathbf{B}}_{\text{out}}$ of dimension $bs \times D \times H$.

end

Transposition of the batch: $\tilde{\mathbf{B}}_{\text{out}}$ is reshaped in dimension $bs \times H \times D$.

RevIN denormalization: $\tilde{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}}$ following Eq. (A.3).

Output: Batch of bs prediction sequences $\hat{\mathbf{Y}} \in \mathbb{R}^{D \times H}$ arranged in a tensor $\hat{\mathbf{B}}_{\text{out}}$ of dimension $bs \times H \times D$.

Training parameters. For all of our experiments, we train our baselines (SAMformer, Transformer, TSMixer with SAM, TSMixer without SAM) with the Adam optimizer (Kingma & Ba, 2015), a batch size of 32, a cosine annealing scheduler (Loshchilov & Hutter, 2017) and the learning rates summarized in Table 3.1. For SAMformer and TSMixer trained with SAM, the values of neighborhood size ρ^* used are reported in Table 3.2. The training/validation/test split is 12/4/4 months on the ETT datasets and 70%/20%/10% on

the other datasets. We use a look-back window $L = 512$ and use a sliding window with stride 1 to create the sequences. The training loss is the MSE on the multivariate time series (Eq. (3.1)). Training is performed during 300 epochs and we use early stopping with a patience of 5 epochs. For each dataset, baselines, and prediction horizon $H \in \{96, 192, 336, 720\}$, each experiment is run 5 times with different seeds, and we display the average and the standard deviation of the test MSE and MAE over the 5 trials.

Table 3.1: Learning rates used in our experiments. ETT designs ETTh1, ETTh2, ETTm1 and ETTm2.

Dataset	ETT	Electricity	Exchange	Traffic	Weather
Learning rate	0.001	0.0001	0.001	0.0001	0.0001

Datasets. We conduct our experiments on 8 publicly available real-world multivariate time series datasets, widely adopted for evaluating long-term forecasting methods (H. Wu et al., 2021; S.-A. Chen et al., 2023; Yuqi Nie et al., 2023; Zeng, M. Chen, et al., 2023). The four Electricity Transformer Temperature datasets ETTh1, ETTh2, ETTm1, and ETTm2 (H. Zhou et al., 2021), collectively referred to as ETT whenever possible, contain measurements from electricity transformers collected between July 2016 and July 2018. Electricity (UCI, 2015) records electricity consumption of 321 clients from 2012 to 2014. Exchange (Guan Lai et al., 2018) includes daily exchange rates between 8 countries from 1990 to 2016. Traffic (California Department of Transportation, 2021) consists of road occupancy rates collected by 862 sensors from January 2015 to December 2016. Lastly, Weather (Max Planck Institute, 2021) gathers meteorological data from 21 weather indicators throughout 2020. Note that Electricity, Traffic, and Weather are large-scale datasets. All time series are segmented into windows of input length $L = 512$ with prediction horizons $H \in \{96, 192, 336, 720\}$ and a stride of 1, meaning each subsequent window shifts by one step. The ETT datasets are available here, and the other four datasets can be found here. Table 3.3 summarizes the main characteristics of these datasets.

Baselines. We compare SAMformer with the previously introduced Transformer and TSMixer (S.-A. Chen et al., 2023), a state-of-the-art baseline built entirely on MLPs. While S.-A. Chen et al. 2023 originally reported results for TSMixer using a single seed, we provide results averaged over multiple runs with different seeds, ensuring a more reliable evaluation. Additionally, for fair comparison, we evaluate TSMixer trained with SAM. We also include results reported by Yong Liu et al. 2024 and S.-A. Chen et al. 2023 for several recent transformer-based baselines: iTransformer (Yong Liu et al., 2024), PatchTST (Yuqi Nie et al., 2023), FEDformer (T. Zhou, Ma, et al., 2022), Informer (H. Zhou et al., 2021), and Autoformer (H. Wu et al., 2021). All models use RevIN (T. Kim et al., 2021), unless explicitly stated otherwise, to maintain consistency.

As described above, all experiments are conducted using a look-back window of length $L = 512$ and prediction horizons $H \in \{96, 192, 336, 720\}$. The results presented in Ta-

Table 3.2: Neighborhood size ρ^* at which SAMformer and TSMixer achieve their best performance on the benchmarks.

H	Model	ETTh1	ETTh2	ETTm1	ETTm2	Electricity	Exchange	Traffic	Weather
96	SAMformer	0.5	0.5	0.6	0.2	0.5	0.7	0.8	0.4
	TSMixer	1.0	0.9	1.0	1.0	0.9	1.0	0.0	0.5
192	SAMformer	0.6	0.8	0.9	0.9	0.6	0.8	0.1	0.4
	TSMixer	0.7	0.1	0.6	1.0	1.0	0.0	0.9	0.4
336	SAMformer	0.9	0.6	0.9	0.8	0.5	0.5	0.5	0.6
	TSMixer	0.7	0.0	0.7	1.0	0.4	1.0	0.6	0.6
720	SAMformer	0.9	0.8	0.9	0.9	1.0	0.9	0.7	0.5
	TSMixer	0.3	0.4	0.5	1.0	0.9	0.1	0.9	0.3

Table 3.3: Characteristics of the multivariate time series datasets used in our experiments with various sizes and dimensions.

Dataset	ETTh1/ETTh2	ETTm1/ETTm2	Electricity	Exchange	Traffic	Weather
# features	7	7	321	8	862	21
# time steps	17420	69680	26304	7588	17544	52696
Granularity	1 hour	15 minutes	1 hour	1 day	1 hour	10 minutes

ble 3.4 for SAMformer, TSMixer, and Transformer are obtained from our own experiments, averaged over 5 runs with different random seeds. Minor differences may be observed between our results for TSMixer without SAM and those reported in S.-A. Chen et al. 2023, since we average performance across multiple seeds for robustness, whereas the original paper reported single-seed performance. We also perform a Student's t-test (Table 3.6) to provide statistical significance between SAMformer and TSMixer trained with SAM.

It is noteworthy that, unlike other baselines including TSMixer, the overall structure of SAMformer remains unchanged across all datasets, demonstrating robustness and eliminating extensive hyperparameter tuning.

Finally, additional baseline results from the literature are included for thorough comparison. Results for Informer, Autoformer, and FEDformer on all datasets except Exchange are sourced from S.-A. Chen et al. 2023. The Exchange dataset results for these models are taken from their original papers and therefore do not use RevIN. Results for iTransformer and PatchTST, reported by Yong Liu et al. 2024, employ RevIN. Note that iTransformer incorporates both temporal and channel-wise attention. Our extensive experimental evaluation thus ensures a comprehensive and fair comparison across leading models in multivariate long-term time series forecasting.

Evaluation. All models are trained to minimize the MSE loss defined in Eq. (3.1). The average MSE on the test set, together with the standard deviation over 5 runs with different

seeds is reported. Additional details and results, including the Mean Absolute Error (MAE), can be found in Table 3.5. Except specified otherwise, all our results are also obtained over 5 runs with different seeds.

3.3.1 Main Takeaways

SAMformer improves over state-of-the-art. The experimental results are detailed in Table 3.4, with a Student's t-test analysis available in Table 3.6. SAMformer outperforms its competitors on **7 out of 8** datasets by a large margin. In particular, it improves over its best competitor TSMixer+SAM by **5.25**%, surpasses the standalone TSMixer by **14.33**% and the best multivariate transformer-based model FEDformer by **12.36**%. In addition, it improves over Transformer by **16.96**%. SAMformer also outperforms the very recent iTransformer, a transformer-based approach that uses both temporal and spatial attention, and PatchTST which was tailored for univariate time series forecasting. We notice that iTransformer has mixed global performance and gets beaten by SAMformer on all datasets, except Exchange on which it significantly outperforms all competitors. This explains that SAMformer improves it only by **3.94**% overall but up to **8.38**% without it. Finally, SAMformer outperforms PatchTST by **11.13**%. For every horizon and dataset (except Exchange), SAMformer is ranked either first or second. Notably, SAM's integration improves the generalization capacity of TSMixer, resulting in an average enhancement of 9.58%. A similar study with the MAE in Table 3.5 leads to the same conclusions. As TSMixer trained with SAM is the second-best baseline almost always ranked second, it serves as a primary benchmark for further discussion in this section. It should be noted that SAMformer has 4 times fewer parameters than TSMixer, and several orders of magnitude fewer than the transformer-based methods.

Significance Test for SAMformer and TSMixer with SAM. In this section, we perform a Student t-test between SAMformer and TSMixer trained with SAM. It should be noted that TSMixer with SAM significantly outperforms vanilla TSMixer. We report the results in Table 3.6. We observe that the SAMformer significantly improves upon TSMixer trained with SAM on 7 out of 8 datasets.

Smoother loss landscape. The introduction of SAM in the training of SAMformer makes its loss smoother than that of Transformer. We illustrate this in Figure 3.6 by comparing the values of $\lambda_{\max}$ for Transformer and SAMformer after training on ETTh1 and Exchange. Our observations reveal that Transformer exhibits considerably higher sharpness, while SAMformer has a desired behavior with a loss landscape sharpness that is an order of magnitude smaller.

Strong Generalization Regardless of the Initialization SAMformer has a strong generalization capacity. In particular, Transformer heavily depends on the initialization, which

Table 3.4: Performance comparison between our model (**SAMformer**) and baselines for multivariate long-term forecasting with different horizons H. Results marked with † are obtained from Yong Liu et al. 2024 and those marked with * are obtained from S.-A. Chen et al. 2023, along with the publication year of the respective methods. Transformer-based models are abbreviated by removing the "former" part of their name. We display the average test MSE with standard deviation obtained on 5 runs with different seeds. **Best** results are in bold, second best are underlined.

Dataset	H	with SAM		without SAM						
		SAMformer	TSMixer	Transformer	TSMixer	iTrans†	PatchTST†	In*	Auto*	FED*
		-	-	-	2023	2024	2023	2021	2021	2022
ETTh1	96	0.381$_{\pm 0.003}$	0.388$_{\pm 0.001}$	0.509$_{\pm 0.031}$	0.398$_{\pm 0.001}$	0.386	0.414	0.941	0.435	**0.376**
	192	**0.409**$_{\pm 0.002}$	0.421$_{\pm 0.002}$	0.535$_{\pm 0.043}$	0.426$_{\pm 0.003}$	0.441	0.460	1.007	0.456	0.423
	336	**0.423**$_{\pm 0.001}$	0.430$_{\pm 0.002}$	0.570$_{\pm 0.016}$	0.435$_{\pm 0.003}$	0.487	0.501	1.038	0.486	0.444
	720	**0.427**$_{\pm 0.002}$	0.440$_{\pm 0.005}$	0.601$_{\pm 0.036}$	0.498$_{\pm 0.076}$	0.503	0.500	1.144	0.515	0.469
ETTh2	96	**0.295**$_{\pm 0.002}$	0.305$_{\pm 0.007}$	0.396$_{\pm 0.017}$	0.308$_{\pm 0.003}$	0.297	0.302	1.549	0.332	0.332
	192	**0.340**$_{\pm 0.002}$	0.350$_{\pm 0.002}$	0.413$_{\pm 0.010}$	0.352$_{\pm 0.004}$	0.380	0.388	3.792	0.426	0.407
	336	**0.350**$_{\pm 0.000}$	0.360$_{\pm 0.002}$	0.414$_{\pm 0.002}$	0.360$_{\pm 0.002}$	0.428	0.426	4.215	0.477	0.400
	720	**0.391**$_{\pm 0.001}$	0.402$_{\pm 0.002}$	0.424$_{\pm 0.009}$	0.409$_{\pm 0.006}$	0.427	0.431	3.656	0.453	0.412
ETTm1	96	0.329$_{\pm 0.001}$	0.327$_{\pm 0.002}$	0.384$_{\pm 0.022}$	0.336$_{\pm 0.004}$	0.334	0.329	0.626	0.510	**0.326**
	192	**0.353**$_{\pm 0.006}$	0.356$_{\pm 0.004}$	0.400$_{\pm 0.026}$	0.362$_{\pm 0.006}$	0.377	0.367	0.725	0.514	0.365
	336	**0.382**$_{\pm 0.001}$	0.387$_{\pm 0.004}$	0.461$_{\pm 0.017}$	0.391$_{\pm 0.003}$	0.426	0.399	1.005	0.510	0.392
	720	**0.429**$_{\pm 0.000}$	0.441$_{\pm 0.002}$	0.463$_{\pm 0.046}$	0.450$_{\pm 0.006}$	0.491	0.454	1.133	0.527	0.446
ETTm2	96	0.181$_{\pm 0.005}$	0.190$_{\pm 0.003}$	0.200$_{\pm 0.036}$	0.211$_{\pm 0.014}$	**0.180**	0.175	0.355	0.205	**0.180**
	192	**0.233**$_{\pm 0.002}$	0.250$_{\pm 0.002}$	0.273$_{\pm 0.013}$	0.252$_{\pm 0.005}$	0.250	0.241	0.595	0.278	0.252
	336	**0.285**$_{\pm 0.001}$	0.301$_{\pm 0.003}$	0.310$_{\pm 0.022}$	0.303$_{\pm 0.004}$	0.311	0.305	1.270	0.343	0.324
	720	**0.375**$_{\pm 0.001}$	0.389$_{\pm 0.002}$	0.426$_{\pm 0.025}$	0.390$_{\pm 0.003}$	0.412	0.402	3.001	0.414	0.410
Electricity	96	**0.155**$_{\pm 0.002}$	0.171$_{\pm 0.001}$	0.182$_{\pm 0.006}$	0.173$_{\pm 0.004}$	-	-	0.304	0.196	0.186
	192	**0.168**$_{\pm 0.001}$	0.191$_{\pm 0.010}$	0.202$_{\pm 0.041}$	0.204$_{\pm 0.027}$	-	-	0.327	0.211	0.197
	336	**0.183**$_{\pm 0.000}$	0.198$_{\pm 0.006}$	0.212$_{\pm 0.017}$	0.217$_{\pm 0.018}$	-	-	0.333	0.214	0.213
	720	**0.219**$_{\pm 0.000}$	0.230$_{\pm 0.005}$	0.238$_{\pm 0.016}$	0.242$_{\pm 0.015}$	-	-	0.351	0.236	0.233
Exchange	96	0.161$_{\pm 0.007}$	0.233$_{\pm 0.016}$	0.292$_{\pm 0.045}$	0.343$_{\pm 0.082}$	**0.086**	0.088	0.847	0.197	0.139
	192	0.246$_{\pm 0.009}$	0.342$_{\pm 0.031}$	0.372$_{\pm 0.035}$	0.342$_{\pm 0.031}$	0.177	**0.176**	1.204	0.300	0.256
	336	0.368$_{\pm 0.006}$	0.474$_{\pm 0.014}$	0.494$_{\pm 0.033}$	0.484$_{\pm 0.062}$	0.331	**0.301**	1.672	0.509	0.426
	720	1.003$_{\pm 0.018}$	1.078$_{\pm 0.179}$	1.323$_{\pm 0.192}$	1.204$_{\pm 0.028}$	**0.847**	0.901	2.478	1.447	1.090
Traffic	96	0.407$_{\pm 0.001}$	0.409$_{\pm 0.016}$	0.420$_{\pm 0.041}$	0.409$_{\pm 0.016}$	**0.395**	0.462	0.733	0.597	0.576
	192	**0.415**$_{\pm 0.005}$	0.433$_{\pm 0.009}$	0.441$_{\pm 0.039}$	0.637$_{\pm 0.444}$	0.417	0.466	0.777	0.607	0.610
	336	**0.421**$_{\pm 0.001}$	0.424$_{\pm 0.000}$	0.501$_{\pm 0.154}$	0.747$_{\pm 0.277}$	0.433	0.482	0.776	0.623	0.608
	720	**0.456**$_{\pm 0.003}$	0.488$_{\pm 0.028}$	0.468$_{\pm 0.021}$	0.688$_{\pm 0.287}$	0.467	0.514	0.827	0.639	0.621
Weather	96	0.197$_{\pm 0.001}$	**0.189**$_{\pm 0.003}$	0.227$_{\pm 0.012}$	0.214$_{\pm 0.004}$	0.174	0.177	0.354	0.249	0.238
	192	0.235$_{\pm 0.000}$	**0.228**$_{\pm 0.004}$	0.256$_{\pm 0.018}$	0.231$_{\pm 0.003}$	0.221	0.225	0.419	0.325	0.275
	336	0.276$_{\pm 0.001}$	**0.271**$_{\pm 0.001}$	0.278$_{\pm 0.001}$	0.279$_{\pm 0.007}$	0.278	0.278	0.583	0.351	0.339
	720	0.334$_{\pm 0.000}$	**0.331**$_{\pm 0.001}$	0.353$_{\pm 0.002}$	0.343$_{\pm 0.024}$	0.358	0.354	0.916	0.415	0.389
Overall MSE improvement		**5.25%**		**16.96%**	**14.33%**	**3.94%**	**11.13%**	**72.20%**	**22.65%**	**12.36%**

might be due to bad local minima as its loss landscape is sharper than the one of SAMformer. We display in Figure 3.7 and Figure 3.8 the distribution of the test MSE on 5 runs on the datasets used in our experiments (Table 3.3) and various prediction horizons

Table 3.5: Performance comparison between our model (**SAMformer**) and baselines for multivariate long-term forecasting with different horizons H. Results marked with † are obtained from Yong Liu et al. 2024 and those marked with * are obtained from S.-A. Chen et al. 2023, along with the publication year of the respective methods. Transformer-based models are abbreviated by removing the "former" part of their name. We display the average test MAE with standard deviation obtained on 5 runs with different seeds. **Best** results are in bold, second best are underlined.

Dataset	H	with SAM		without SAM						
		SAMformer	TSMixer	Transformer	TSMixer	iTrans†	PatchTST†	In*	Auto*	FED*
		-	-	-	2023	2024	2023	2021	2021	2022
ETTh1	96	**0.402**$_{\pm 0.001}$	0.408$_{\pm 0.001}$	0.619$_{\pm 0.203}$	0.414$_{\pm 0.004}$	0.405	0.419	0.769	0.446	0.415
	192	**0.418**$_{\pm 0.001}$	0.426$_{\pm 0.002}$	0.513$_{\pm 0.024}$	0.428$_{\pm 0.001}$	0.436	0.445	0.786	0.457	0.446
	336	**0.425**$_{\pm 0.000}$	0.434$_{\pm 0.001}$	0.529$_{\pm 0.008}$	0.434$_{\pm 0.001}$	0.458	0.466	0.784	0.487	0.462
	720	**0.449**$_{\pm 0.002}$	0.459$_{\pm 0.004}$	0.553$_{\pm 0.021}$	0.506$_{\pm 0.064}$	0.491	0.488	0.857	0.517	0.492
ETTh2	96	0.358$_{\pm 0.002}$	0.367$_{\pm 0.002}$	0.416$_{\pm 0.025}$	0.367$_{\pm 0.003}$	0.349	**0.348**	0.952	0.368	0.374
	192	**0.386**$_{\pm 0.003}$	0.393$_{\pm 0.001}$	0.435$_{\pm 0.019}$	0.395$_{\pm 0.003}$	0.400	0.400	1.542	0.434	0.446
	336	**0.395**$_{\pm 0.002}$	0.404$_{\pm 0.004}$	0.434$_{\pm 0.014}$	0.404$_{\pm 0.002}$	0.432	0.433	1.642	0.479	0.447
	720	**0.428**$_{\pm 0.001}$	0.435$_{\pm 0.002}$	0.448$_{\pm 0.006}$	0.441$_{\pm 0.005}$	0.445	0.446	1.619	0.490	0.469
ETTm1	96	**0.363**$_{\pm 0.001}$	**0.363**$_{\pm 0.001}$	0.395$_{\pm 0.024}$	0.371$_{\pm 0.002}$	0.368	0.367	0.560	0.492	0.390
	192	**0.378**$_{\pm 0.003}$	0.381$_{\pm 0.002}$	0.414$_{\pm 0.027}$	0.384$_{\pm 0.003}$	0.391	0.385	0.619	0.495	0.415
	336	**0.394**$_{\pm 0.001}$	0.397$_{\pm 0.002}$	0.445$_{\pm 0.009}$	0.399$_{\pm 0.003}$	0.420	0.410	0.741	0.492	0.425
	720	**0.418**$_{\pm 0.000}$	0.425$_{\pm 0.001}$	0.456$_{\pm 0.035}$	0.429$_{\pm 0.002}$	0.459	0.439	0.845	0.493	0.458
ETTm2	96	0.274$_{\pm 0.010}$	0.284$_{\pm 0.004}$	0.290$_{\pm 0.026}$	0.302$_{\pm 0.013}$	0.264	**0.259**	0.462	0.293	0.271
	192	0.306$_{\pm 0.001}$	0.320$_{\pm 0.001}$	0.347$_{\pm 0.025}$	0.323$_{\pm 0.005}$	0.309	**0.302**	0.586	0.336	0.318
	336	**0.338**$_{\pm 0.001}$	0.350$_{\pm 0.001}$	0.360$_{\pm 0.017}$	0.352$_{\pm 0.003}$	0.348	0.343	0.871	0.379	0.364
	720	**0.390**$_{\pm 0.001}$	0.402$_{\pm 0.002}$	0.424$_{\pm 0.014}$	0.402$_{\pm 0.003}$	0.407	0.400	1.267	0.419	0.420
Electricity	96	**0.252**$_{\pm 0.002}$	0.273$_{\pm 0.001}$	0.288$_{\pm 0.013}$	0.277$_{\pm 0.003}$	-	-	0.393	0.313	0.302
	192	**0.263**$_{\pm 0.001}$	0.292$_{\pm 0.011}$	0.304$_{\pm 0.033}$	0.304$_{\pm 0.027}$	-	-	0.417	0.324	0.311
	336	**0.277**$_{\pm 0.000}$	0.297$_{\pm 0.007}$	0.315$_{\pm 0.018}$	0.317$_{\pm 0.018}$	-	-	0.422	0.327	0.328
	720	**0.306**$_{\pm 0.000}$	0.321$_{\pm 0.006}$	0.330$_{\pm 0.014}$	0.333$_{\pm 0.015}$	-	-	0.427	0.342	0.344
Exchange	96	0.306$_{\pm 0.006}$	0.363$_{\pm 0.013}$	0.369$_{\pm 0.049}$	0.436$_{\pm 0.054}$	0.206	**0.205**	0.752	0.323	0.276
	192	0.371$_{\pm 0.008}$	0.437$_{\pm 0.021}$	0.416$_{\pm 0.041}$	0.437$_{\pm 0.021}$	**0.299**	**0.299**	0.895	0.369	0.369
	336	0.453$_{\pm 0.004}$	0.515$_{\pm 0.006}$	0.491$_{\pm 0.036}$	0.523$_{\pm 0.029}$	0.417	**0.397**	1.036	0.524	0.464
	720	0.750$_{\pm 0.006}$	0.777$_{\pm 0.064}$	0.823$_{\pm 0.040}$	0.818$_{\pm 0.007}$	**0.691**	0.714	1.310	0.941	0.800
Traffic	96	0.292$_{\pm 0.001}$	0.300$_{\pm 0.020}$	0.306$_{\pm 0.033}$	0.300$_{\pm 0.020}$	**0.268**	0.295	0.410	0.371	0.359
	192	0.294$_{\pm 0.005}$	0.317$_{\pm 0.012}$	0.321$_{\pm 0.034}$	0.419$_{\pm 0.218}$	**0.276**	0.296	0.435	0.382	0.380
	336	0.292$_{\pm 0.000}$	0.299$_{\pm 0.000}$	0.348$_{\pm 0.093}$	0.501$_{\pm 0.163}$	**0.283**	0.304	0.434	0.387	0.375
	720	0.311$_{\pm 0.003}$	0.344$_{\pm 0.026}$	0.325$_{\pm 0.023}$	0.458$_{\pm 0.159}$	**0.302**	0.322	0.466	0.395	0.375
Weather	96	0.249$_{\pm 0.001}$	0.242$_{\pm 0.002}$	0.281$_{\pm 0.018}$	0.271$_{\pm 0.009}$	**0.214**	0.218	0.405	0.329	0.314
	192	0.277$_{\pm 0.000}$	0.272$_{\pm 0.003}$	0.302$_{\pm 0.020}$	0.275$_{\pm 0.003}$	**0.254**	0.259	0.434	0.370	0.329
	336	0.304$_{\pm 0.001}$	0.299$_{\pm 0.001}$	0.310$_{\pm 0.012}$	0.307$_{\pm 0.009}$	**0.296**	0.297	0.543	0.391	0.377
	720	0.342$_{\pm 0.000}$	**0.341**$_{\pm 0.002}$	0.363$_{\pm 0.002}$	0.351$_{\pm 0.021}$	0.347	0.348	0.705	0.426	0.409
Overall MAE improvement		**3.99%**		**11.63%**	**9.60%**	**2.05%**	**2.75%**	**53.00%**	**15.67%**	**9.93%**

$H \in \{96, 192, 336, 720\}$. SAMformer consistently demonstrates strong and stable performance across different datasets and horizons, independent of the seed. On the contrary,

Table 3.6: Significance test with Student's t-test and performance comparison between SAMformer and TSMixer trained with SAM across various datasets and prediction horizons. We display the average and standard deviation of the test MSE obtained on 5 runs ($\text{mean}_{\pm\text{std}}$). The performance of the best model is in **bold** when the improvement is statistically significant at the level 0.05 (p-value < 0.05).

H	Model	ETTh1	ETTh2	ETTm1	ETTm2	Electricity	Exchange	Traffic	Weather
96	SAMformer	**0.381**$_{\pm 0.003}$	**0.295**$_{\pm 0.002}$	0.329$_{\pm 0.001}$	**0.181**$_{\pm 0.005}$	**0.155**$_{\pm 0.002}$	**0.161**$_{\pm 0.007}$	0.407$_{\pm 0.001}$	0.197$_{\pm 0.001}$
	TSMixer	0.388$_{\pm 0.001}$	0.305$_{\pm 0.007}$	0.327$_{\pm 0.002}$	0.190$_{\pm 0.003}$	0.171$_{\pm 0.001}$	0.233$_{\pm 0.016}$	0.409$_{\pm 0.016}$	**0.189**$_{\pm 0.003}$
192	SAMformer	**0.409**$_{\pm 0.002}$	**0.340**$_{\pm 0.002}$	0.353$_{\pm 0.006}$	**0.233**$_{\pm 0.002}$	**0.168**$_{\pm 0.001}$	**0.246**$_{\pm 0.009}$	**0.415**$_{\pm 0.005}$	0.235$_{\pm 0.000}$
	TSMixer	0.421$_{\pm 0.002}$	0.350$_{\pm 0.002}$	0.356$_{\pm 0.004}$	0.250$_{\pm 0.002}$	0.191$_{\pm 0.010}$	0.342$_{\pm 0.031}$	0.433$_{\pm 0.009}$	**0.228**$_{\pm 0.004}$
336	SAMformer	**0.423**$_{\pm 0.001}$	**0.350**$_{\pm 0.000}$	**0.382**$_{\pm 0.001}$	**0.285**$_{\pm 0.001}$	**0.183**$_{\pm 0.000}$	**0.368**$_{\pm 0.006}$	**0.421**$_{\pm 0.001}$	0.276$_{\pm 0.001}$
	TSMixer	0.430$_{\pm 0.002}$	0.360$_{\pm 0.002}$	0.387$_{\pm 0.004}$	0.301$_{\pm 0.003}$	0.198$_{\pm 0.006}$	0.474$_{\pm 0.014}$	0.424$_{\pm 0.000}$	**0.271**$_{\pm 0.001}$
720	SAMformer	**0.427**$_{\pm 0.002}$	**0.391**$_{\pm 0.001}$	**0.429**$_{\pm 0.000}$	**0.375**$_{\pm 0.001}$	**0.219**$_{\pm 0.000}$	1.003$_{\pm 0.018}$	**0.456**$_{\pm 0.003}$	0.334$_{\pm 0.000}$
	TSMixer	0.440$_{\pm 0.005}$	0.402$_{\pm 0.002}$	0.441$_{\pm 0.002}$	0.389$_{\pm 0.002}$	0.230$_{\pm 0.005}$	1.078$_{\pm 0.179}$	0.488$_{\pm 0.028}$	**0.331**$_{\pm 0.001}$

the performance of Transformer is unstable with a large generalization gap depending on the seed.

3.3.2 Qualitative Benefits of Our Approach

Computational efficiency. SAMformer is computationally more efficient than TSMixer and usual transformer-based approaches, benefiting from a shallow lightweight implementation, i.e., a single layer with one attention head. The number of parameters of SAMformer and TSMixer is detailed in Table 3.7. We observe that, on average, SAMformer has ~ 4 times fewer parameters than TSMixer, which makes this approach even more remarkable. Importantly, TSMixer itself is recognized as a computationally efficient architecture compared to the transformer-based baselines (S.-A. Chen et al., 2023, Table 6).

Fewer hyperparameters and versatility. SAMformer requires minimal hyperparameters tuning, contrary to other baselines, including TSMixer and FEDformer. In particular, SAMformer's architecture remains the same for all our experiments, while TSMixer varies in terms of the number of residual blocks and feature embedding dimensions, depending on the dataset. This versatility also comes with better robustness to the prediction horizon H. In Figure 3.9, we display the evolution forecasting accuracy on all datasets for $H \in \{96, 192, 336, 720\}$ for SAMformer and TSMixer (trained with SAM). We observe that SAMformer consistently outperforms its best competitor TSMixer (trained with SAM) for all horizons.

Better attention. We display the attention matrices after training on Weather with the prediction horizon $H = 96$ for Transformer, SAMformer and Transformer + σReparam in Figure 3.10. We note that Transformer excludes self-correlation between features, having low values on the diagonal, while SAMformer strongly promotes them. This pattern is

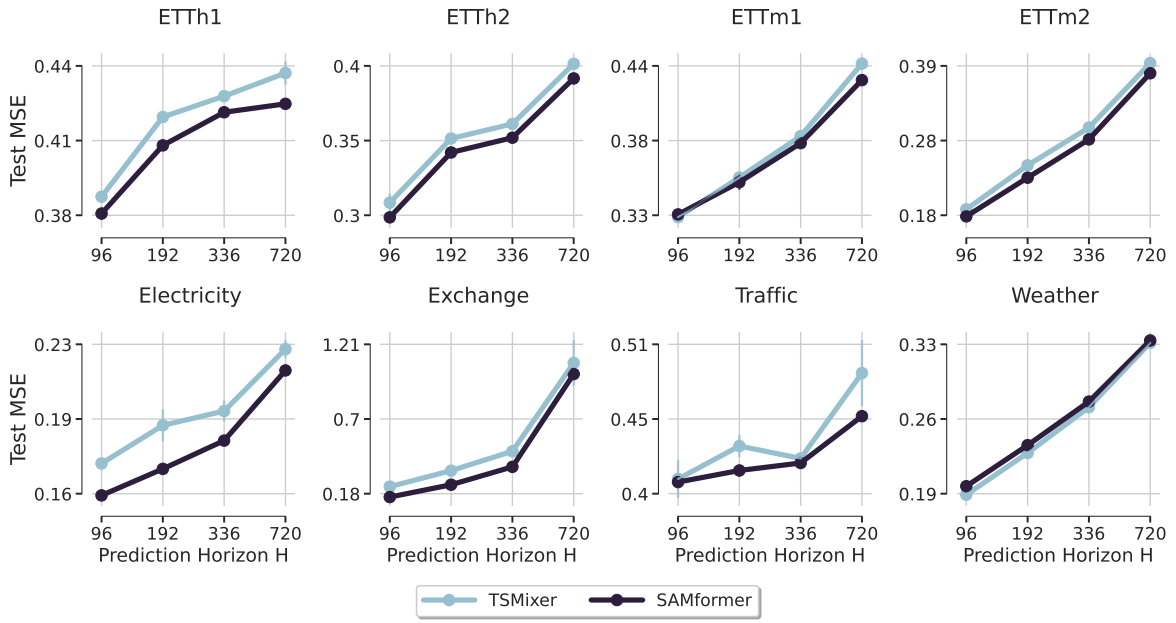


Figure 3.9: Evolution of the test MSE on all datasets for a prediction horizon $H \in \{96, 192, 336, 720\}$. We display the average test MSE with a 95% confidence interval. We see that SAMformer consistently performs well with a low variance. Despite its lightweight (Table 3.7), SAMformer surpasses TSMixer (trained with SAM) on 7 out of 8 datasets as shown in Table 3.4 and Table 3.6.

reminiscent of B. He et al. 2023 and Trockman & Kolter 2023: both works demonstrated the importance of diagonal patterns in attention matrices for signal propagation in transformers used in NLP and computer vision. Our experiments reveal that these insights also apply to time-series forecasting. Note that freezing the attention to $\mathbf{A}(\mathbf{X}) = \mathbf{I}_D$ is largely outperformed by SAMformer as shown in Table 3.10, which confirms the importance of learnable attention. The attention matrix given by σReparam at Figure 3.10 has almost equal rows, leading to rank collapse. In Figure 3.11, we display the distributions of nuclear norms of attention matrices after training Transformer, SAMformer and σReparam. We observe that σReparam heavily penalizes the nuclear norms of the attention matrix, which is coherent with Proposition 3.2.2. In contrast, SAMformer maintains it above Transformer, thus improving the expressiveness of attention.

Computational Efficiency of SAMformer. We compare in Table 3.7 the number of parameters of SAMformer and TSMixer on the several benchmarks used in our experiments. We also display the ratio between the number of parameters of TSMixer and the number of parameters of SAMformer. Overall, SAMformer has ~ 4 times fewer parameters than TSMixer while outperforming it by 14.33% on average.

Faithful Signal Propagation. In this section, we consider Transformer, SAMformer, σReparam, which corresponds to Transformer with the rescaling proposed by Zhai et

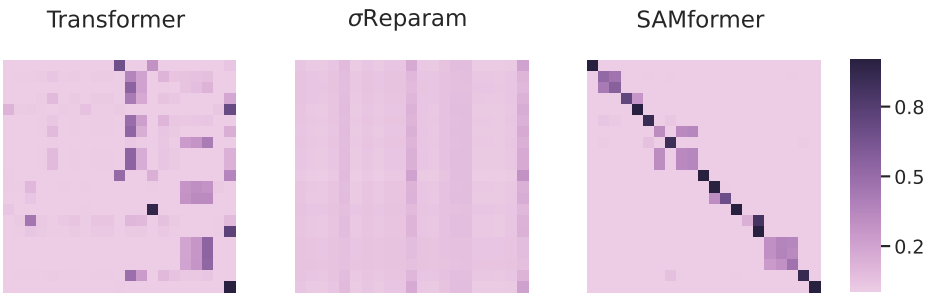


Figure 3.10: Attention matrices on Weather dataset. SAMformer preserves self-correlation among features while σReparam degrades the rank, hindering the propagation of information.

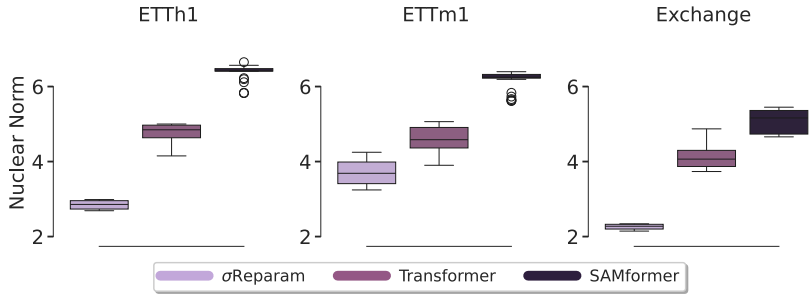
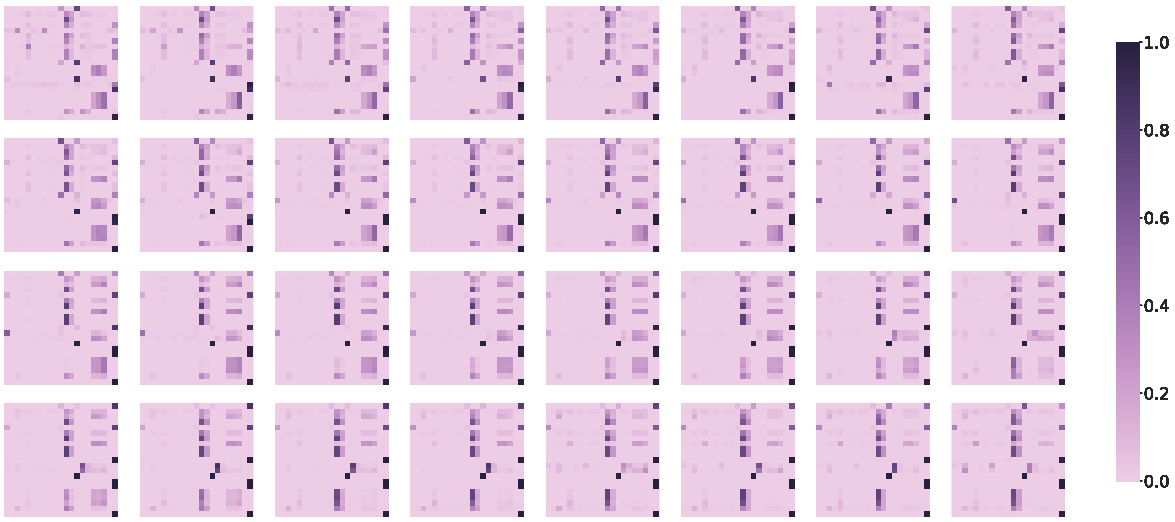


Figure 3.11: Nuclear norm of the attention matrix for different models: σReparam induces lower nuclear norm in accordance with Proposition 3.2.2, while SAMformer keeps the expressiveness of the attention over Transformer.

al. 2023 and SAMformer + σReparam which is SAMformer with the rescaling proposed by Zhai et al. 2023. We plot a batch of attention matrices after training with prediction horizon $H = 96$ (our primary study does not identify significant changes with the value of horizon) on Weather in Figures 3.12 and 3.13. While Transformer tends to ignore the importance of a feature on itself by having low values on the diagonal, we can see in Figure 3.13 that SAMformer strongly encourages these feature-to-feature correlations. A very distinctive pattern is observable: a near-identity attention reminiscent of B. He et al. 2023 and Trockman & Kolter 2023. The former showed that pretrained vision models present similar patterns and both identified the benefits of such attention matrices for the propagation of information along the layers of deep transformers in NLP and computer vision. While in our setting, we have a single-layer transformer, this figure indicates that at the end of the training, self-information from features to themselves is not lost. In contrast, we see that σReparam leads to almost rank-1 matrices with identical columns. This confirms the theoretical insights from Theorem 3.2.2 that showed how rescaling the trainable weights with σReparam to limit the magnitude of $\|\mathbf{W}_Q\mathbf{W}_K^\top\|_2$ could hamper the rank of $\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top$ and of the attention matrix. Finally, we observe that naively combining SAMformer with σReparam does not solve the issues: while some diagonal patterns remain, most of the information has been lost. Moreover, combining both σReparam and SAMformer heavily increases the training time, as shown in Figure 3.14.



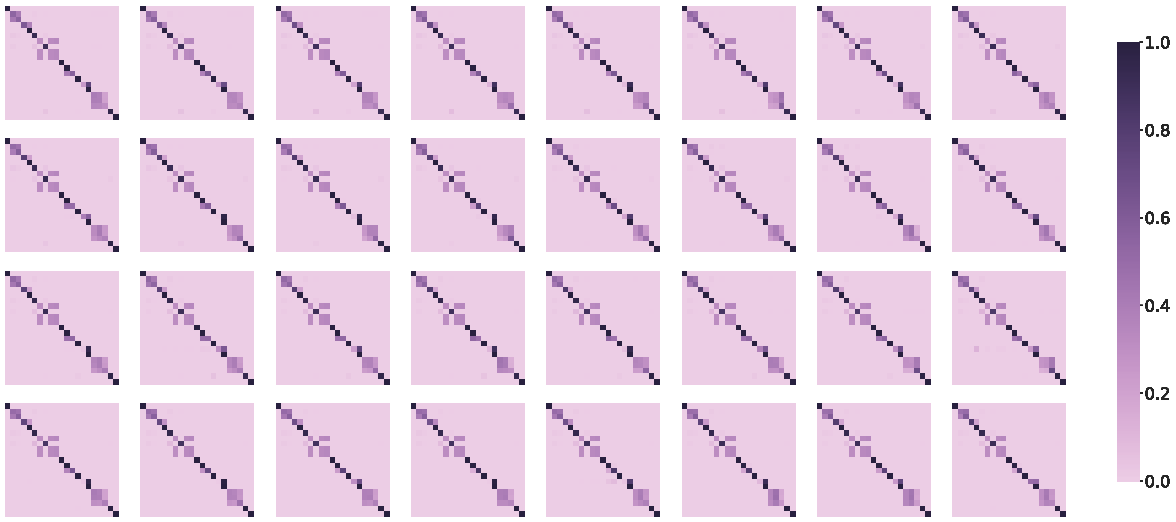
(a) Transformer

(b) σReparam

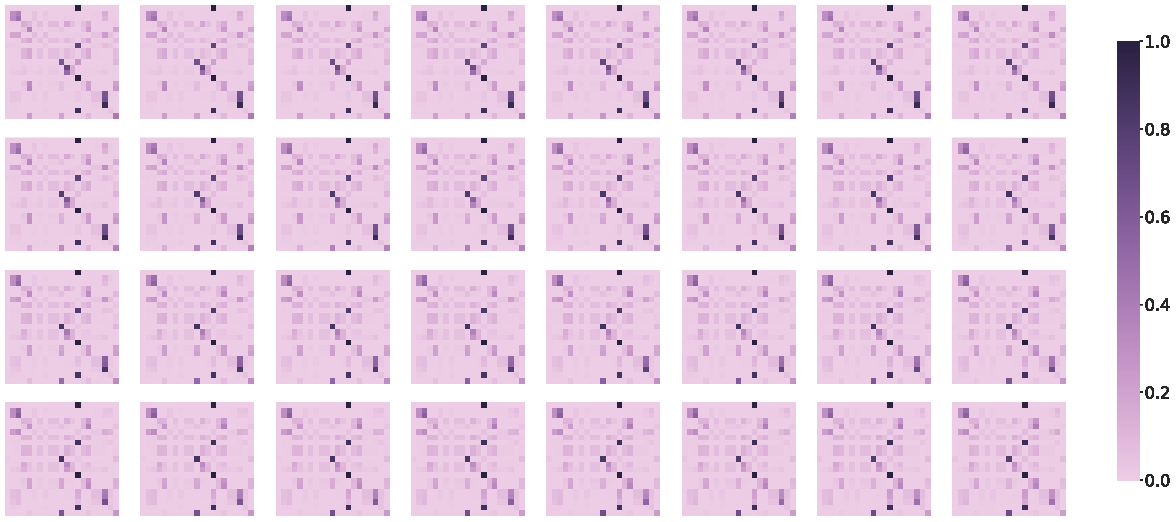
Figure 3.12: Batch of 32 attention matrices on Weather with horizon $H = 96$ after training: (a) Transformer, (b) σReparam.

3.3.3 SAMformer vs MOIRAI

In this section, we show that despite its simplicity, SAMformer is a strong baseline competing not only with the dedicated time series methods (Table 3.4), such as TSMixer but also with the biggest existing time series forecasting foundation model MOIRAI (G. Woo et al., 2024c) that was trained on the largest pretraining corpus LOTSA with nearly **27 billion observations**. MOIRAI was provided in three sizes: small (14 million parameters), base (91 million) and large (314 million). Table 3.8 shows that SAMformer performs on par with MOIRAI on most datasets, surpasses it on three, and overall achieves improvements ranging from at least $\mathbf{1}.\mathbf{1}\%$ to $\mathbf{7}.\mathbf{6}\%$. This comparison highlights again the fact that SAMformer shows impressive performance, globally superior to its competitors while having



(a) SAMformer



(b) SAMformer + σReparam

Figure 3.13: Batch of 32 attention matrices on Weather with horizon $H = 96$ after training: (a) SAMformer, and (b) SAMformer + σReparam.

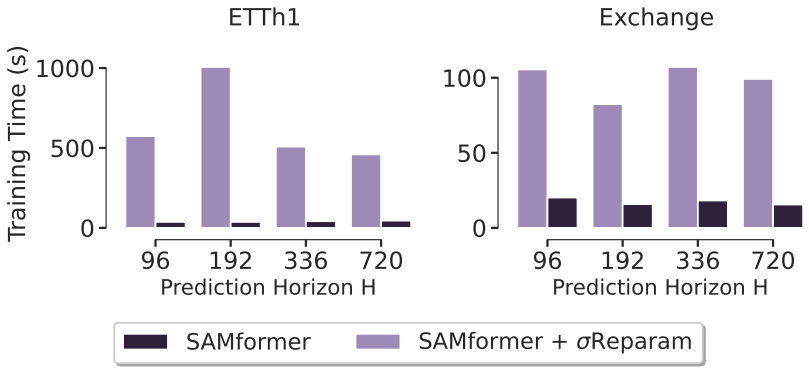


Figure 3.14: Using σReparam on top of SAMformer heavily increases the training time.

Table 3.7: Comparison of the number of parameters between SAMformer and TSMixer on the datasets described in Table 3.3 for prediction horizons $H \in \{96, 192, 336, 720\}$. We also compute the ratio between the number of parameters of TSMixer and the number of parameters of SAMformer. A ratio of 10 means that TSMixer has 10 times more parameters than SAMformer. For each dataset, we display in the last cell of the corresponding row the ratio averaged over all the horizons H. The overall ratio over all datasets and horizons is displayed in **bold** in the bottom right-hand cell.

Dataset	$H = 96$		$H = 192$		$H = 336$		$H = 720$		**Total**
	SAMformer	TSMixer	SAMformer	TSMixer	SAMformer	TSMixer	SAMformer	TSMixer	
ETT	50272	124142	99520	173390	173392	247262	369904	444254	-
Exchange	50272	349344	99520	398592	173392	472464	369904	669456	-
Weather	50272	121908	99520	171156	173392	245028	369904	442020	-
Electricity	50272	280676	99520	329924	173392	403796	369904	600788	-
Traffic	50272	793424	99520	842672	173392	916544	369904	1113536	-
Avg. Ratio	6.64		3.85		2.64		1.77		**3.73**

Table 3.8: Comparison performance of **SAMformer** and MOIRAI (G. Woo et al., 2024c) for multivariate long-term forecasting. We display the test MSE averaged over horizons $\{96, 192, 336, 720\}$. **Best** results are in bold, second best are underlined.

Dataset	Full-shot	Zero-shot (G. Woo et al., 2024c).		
	SAMformer	$MOIRAI_{\text{Small}}$	$MOIRAI_{\text{Base}}$	$MOIRAI_{\text{Large}}$
ETTh1	0.410	**0.400**	0.434	0.510
ETTh2	0.344	**0.341**	0.345	0.354
ETTm1	**0.373**	0.448	0.381	0.390
ETTm2	**0.269**	0.300	0.272	0.276
Electricity	**0.181**	0.233	0.188	0.188
Weather	0.260	0.242	**0.238**	0.259
Overall MSE improvement		**6.9%**	**1.1%**	**7.6%**

much less trainable parameters.

3.3.4 Ablation Study and Sensitivity Analysis

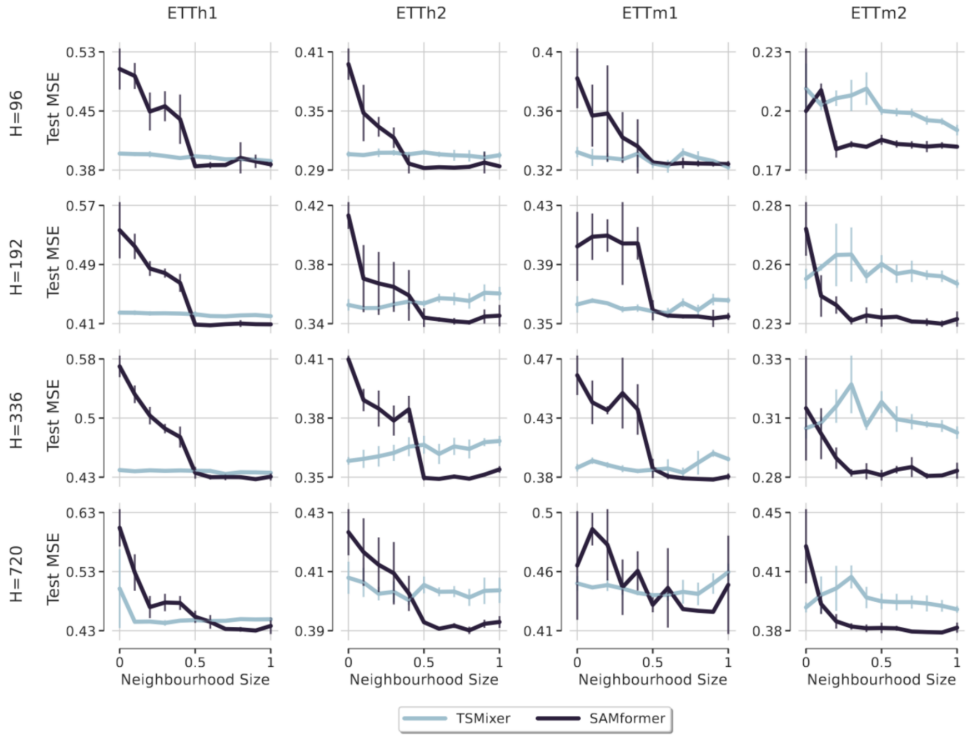
Choices of implementation. We compared our architecture, which utilizes channel-wise attention (Eq.(3.3)), with a temporal-wise attention approach. As shown in Table 3.9, our method demonstrates superior performance in the evaluated setting. Experiments were conducted using Adam (Kingma & Ba, 2015), the standard optimizer for transformers (Ahn et al., 2023; Y. Pan & Y. Li, 2022; T. Zhou, Ma, et al., 2022; H. Zhou et al., 2021; X. Chen et al., 2022). An in-depth ablation study is provided to justify this choice. As expected (Ahn

et al., 2023; L. Liu et al., 2020; Y. Pan & Y. Li, 2022; J. Zhang et al., 2020), SGD (Nesterov, 1983) fails to converge while AdamW (Loshchilov & Hutter, 2019) also struggles, showing unstable performance that is highly sensitive to the choice of weight decay strength, as illustrated in Figure 3.18.

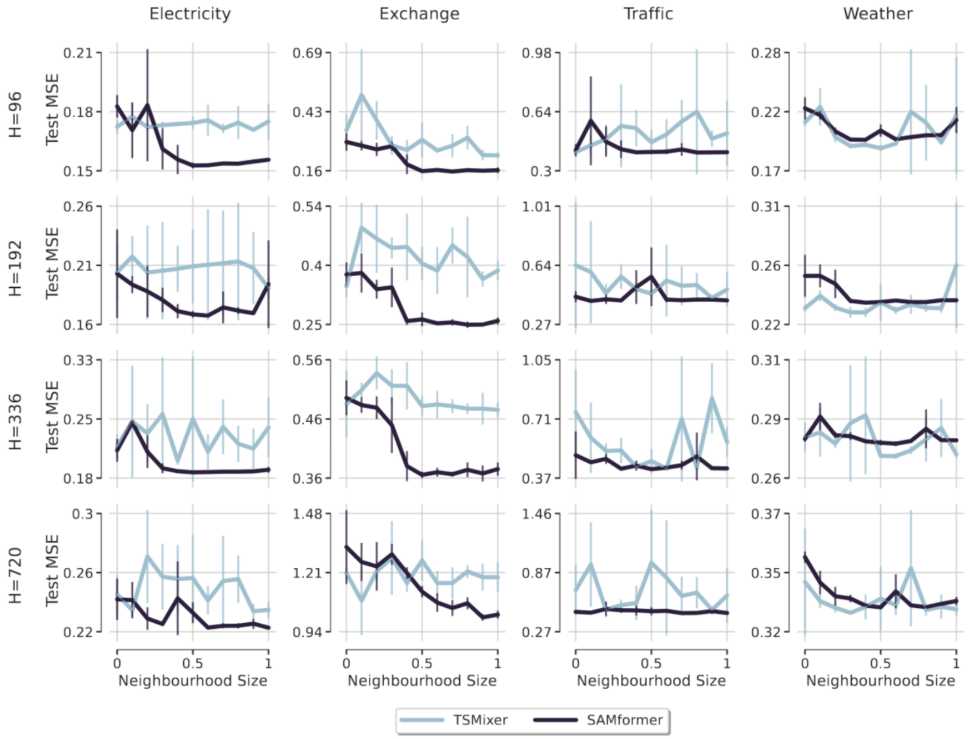
Sensitivity to the neighborhood size ρ. The test MSE of SAMformer and TSMixer is depicted in Figure 3.15 as a function of the neighborhood size ρ. It appears that TSMixer, with its quasi-linear architecture, exhibits less sensitivity to ρ compared to SAMformer. This behavior is consistent with the understanding that, in linear models, the sharpness does not change with respect to ρ, given the constant nature of the loss function's Hessian. Consequently, TSMixer benefits less from changes in ρ than SAMformer. Our observations consistently show that a sufficiently large ρ, generally above 0.7 enables SAMformer to achieve lower MSE than TSMixer.

SAM vs σReparam. We mentioned previously that σReparam doesn't improve the performance of a transformer on a simple toy example, although it makes it comparable to the performance of a transformer with fixed random attention. To further show that σReparam doesn't provide an improvement on real-world datasets, we show in Figure 3.16 that on ETTh1 and Exchange, σReparam alone fails to match SAMformer's improvements, even underperforming Transformer in some cases. A potential improvement may come from combining SAM and σReparam to smooth a rather sparse matrix obtained with SAM. However, as Figure 3.17 illustrates, this combination does not surpass the performance of using SAM alone. Furthermore, combining SAM and σReparam significantly increases training time and memory usage, especially for larger datasets and longer horizons (Figure 3.14), indicating its inefficiency as a method.

Sensitivity to the Change of the Optimizer. In our work, we considered the Adam optimizer (Kingma & Ba, 2015) as it is the de-facto optimizer for transformer-based models (Ahn et al., 2023; Y. Pan & Y. Li, 2022; T. Zhou, Ma, et al., 2022; H. Zhou et al., 2021; X. Chen et al., 2022). The superiority of Adam to optimize networks with attention has been empirically and theoretically studied, where recent works show that the SGD (Nesterov, 1983) was not suitable for attention-based models (Ahn et al., 2023; L. Liu et al., 2020; Y. Pan & Y. Li, 2022; J. Zhang et al., 2020). To ensure the thoroughness of our investigation, we conducted experiments on the synthetic dataset introduced in Eq. (3.2) and reported the results in Figure 3.18a. As expected, we see that using SGD leads to high-magnitude losses and divergence. We also conducted the same experiments with the AdamW (Loshchilov & Hutter, 2019) that incorporates the weight decay scheme in the adaptive optimizer Adam (Kingma & Ba, 2015). We display the results obtained with weight decay factors $\mathsf{wd} = 1\mathrm{e}{-3}$ in Figure 3.18a and with $\mathsf{wd} \in \{1\mathrm{e}{-5}, 1\mathrm{e}{-4}\}$ in Figure 3.18b. When $\mathsf{wd} = 1\mathrm{e}{-3}$, we observe that it does not converge. However, with $\mathsf{wd} \in \{1\mathrm{e}{-5}, 1\mathrm{e}{-4}\}$, we observe a similar behavior for Transformer than when it is trained with Adam (Figure 3.2). Hence, using AdamW does not lead to the significant benefits brought by SAM (Figure 3.1). As the



(a) Sensitivity analysis on ETT datasets.



(b) Sensitivity analysis on Electricity, Exchange, Traffic and Weather datsets.

Figure 3.15: Test MSE vs. ρ (Remark A.1.1), with mean MSE and 95% confidence interval. SAMformer is smoother and generally outperforms TSMixer over wide ρ ranges. For $\rho = 0$, SAM reduces to Adam, confirming consistent improvements. Despite fewer parameters (Table 3.7), SAMformer achieves the lowest MSE on 7/8 datasets (Tables 3.4, 3.6). Compared to X. Chen et al. 2022, larger ρ is needed to improve generalization.

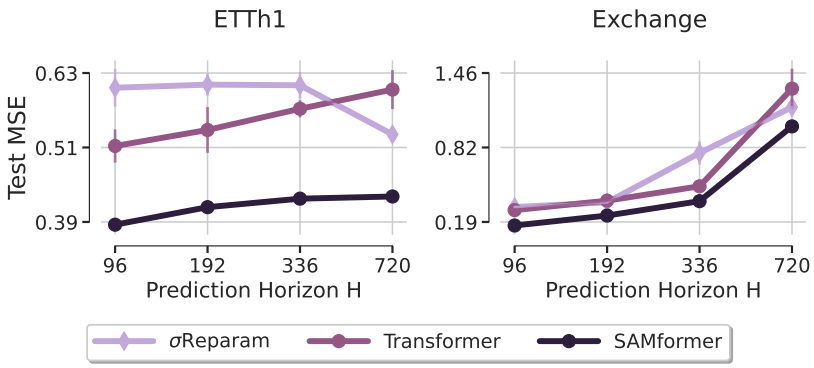


Figure 3.16: Suboptimality of σReparam. Comparison of Transformer, σReparam, and SAMformer. The results indicate that σReparam alone does not improve the performance of Transformer and is clearly outperformed by SAMformer.

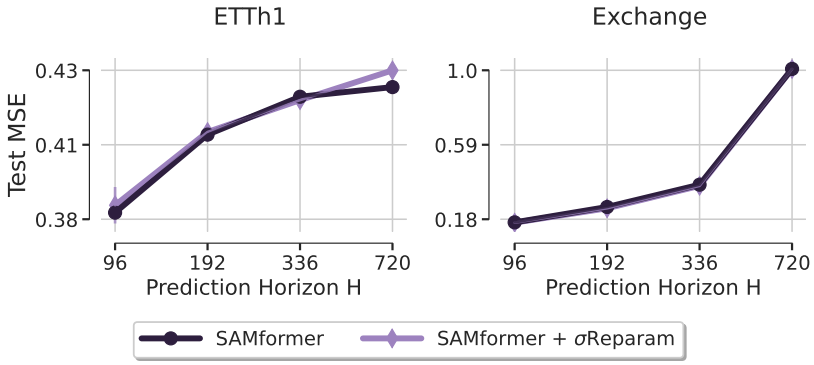
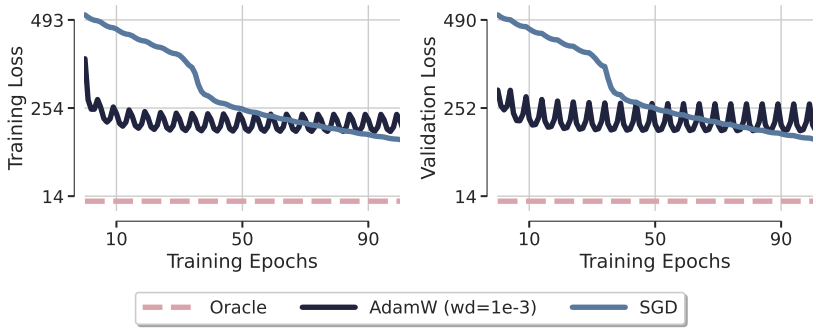


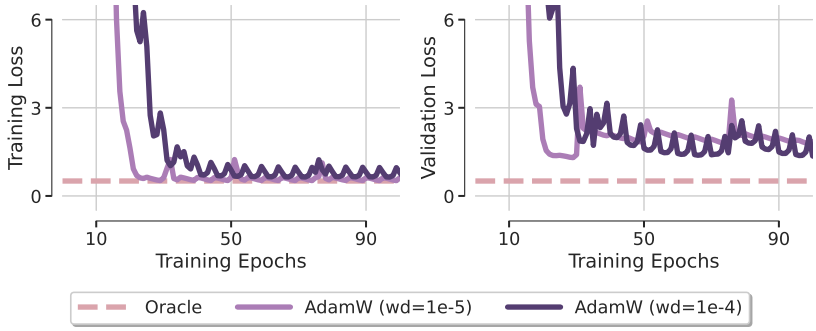
Figure 3.17: Suboptimality of σReparam. Comparison of SAMformer and SAMformer augmented with σReparam. While the combination does not yield a significant improvement in performance, it substantially increases the training time (see Figure 3.14).

optimization is very sensitive to the value of weight decay wd, it motivates us to conduct our experiments with Adam.

Ablation on the Implementation. This ablation study contrasts two variants of our model to showcase the effectiveness of Sharpness-Aware Minimization (SAM) and our attention approach. Identity Attention represents SAMformer with an attention weight matrix constrained to identity, illustrating that SAM does not simply reduce the attention weight matrix to identity, as performance surpasses this configuration. Temporal Attention is compared to our Transformer without SAM, highlighting our focus on treating feature correlations in the attention mechanism rather than temporal correlations.



(a) SGD and AdamW with wd $= 1e{-}3$



(b) AdamW with wd $\in \{1e{-}5, 1e{-}4\}$.

Figure 3.18: Illustration of different optimizers on synthetic data generated with Eq. (3.2) where Oracle is the least-square solution. We saw in Figure 3.1 that with Adam, Transformer overfits and has poor performance while SAMformer smoothly reaches the oracle. (a) We observe that using SGD and Adam with weight decay wd $= 1e{-}5$ leads to huge loss magnitudes and fails to converge. (b) With well-chosen weight decays (wd $\in \{1e{-}3, 1e{-}4\}$), training Transformer with AdamW leads to similar performance than Adam. The overfitting is noticeable and the training is unstable. AdamW does not bring more stabilization and is very sensitive to the hyperparameters. Hence, this toy example motivates us to conduct our thorough experiments with the optimizer Adam.

3.4 Discussion and Future Work

In this work, we demonstrated how simple transformers can reclaim their place as state-of-the-art models in long-term multivariate series forecasting from their MLP-based competitors. Rather than concentrating on new architectures and attention mechanisms, we analyzed the current pitfalls of transformers in this task and addressed them by carefully designing an appropriate training strategy. Our findings suggest that even a simple shallow transformer has a very sharp loss landscape which makes it converge to poor local minima. We analyzed popular solutions proposed in the literature to address this issue and showed which of them work or fail. Our proposed SAMformer, optimized with sharpness-aware minimization, leads to a substantial performance gain compared to the existing forecasting baselines, including the current largest foundation model MOIRAI, and benefits from

Table 3.9: The Temporal Attention model is benchmarked against our Transformer model, which employs feature-based attention rather than time-step-based attention. We report in the last column the **Overall improvement** in MSE and MAE of Transformer over the Temporal Attention. This comparison reveals that channel-wise attention, i.e., focusing on features pairwise correlations, significantly boosts the performance, with a 12.97% improvement in MSE and 18.09% in MAE across all considered datasets.

Model	Metrics	H	ETTh1	ETTh2	ETTm1	ETTm2	Electricity	Exchange	Traffic	Weather	**Overall Improvement**
Temporal Attention	MSE	96	$0.496_{\pm 0.009}$	$0.401_{\pm 0.011}$	$0.542_{\pm 0.063}$	$0.330_{\pm 0.034}$	$0.291_{\pm 0.025}$	$0.684_{\pm 0.218}$	$0.933_{\pm 0.188}$	$0.225_{\pm 0.005}$	12.97%
		192	$0.510_{\pm 0.014}$	$0.414_{\pm 0.020}$	$0.615_{\pm 0.056}$	$0.394_{\pm 0.033}$	$0.294_{\pm 0.024}$	$0.434_{\pm 0.063}$	$0.647_{\pm 0.131}$	$0.254_{\pm 0.001}$	
		336	$0.549_{\pm 0.017}$	$0.396_{\pm 0.014}$	$0.620_{\pm 0.046}$	$0.436_{\pm 0.081}$	$0.290_{\pm 0.016}$	$0.473_{\pm 0.014}$	$0.656_{\pm 0.113}$	$0.292_{\pm 0.000}$	
		720	$0.604_{\pm 0.017}$	$0.396_{\pm 0.010}$	$0.694_{\pm 0.055}$	$0.469_{\pm 0.005}$	$0.307_{\pm 0.014}$	$1.097_{\pm 0.084}$	-	$0.346_{\pm 0.000}$	
	MAE	96	$0.488_{\pm 0.007}$	$0.434_{\pm 0.006}$	$0.525_{\pm 0.040}$	$0.393_{\pm 0.020}$	$0.386_{\pm 0.014}$	$0.589_{\pm 0.096}$	$0.598_{\pm 0.072}$	$0.277_{\pm 0.004}$	18.09%
		192	$0.492_{\pm 0.010}$	$0.443_{\pm 0.015}$	$0.566_{\pm 0.032}$	$0.421_{\pm 0.019}$	$0.385_{\pm 0.014}$	$0.498_{\pm 0.033}$	$0.467_{\pm 0.072}$	$0.294_{\pm 0.001}$	
		336	$0.517_{\pm 0.012}$	$0.440_{\pm 0.012}$	$0.550_{\pm 0.024}$	$0.443_{\pm 0.039}$	$0.383_{\pm 0.009}$	$0.517_{\pm 0.008}$	$0.469_{\pm 0.070}$	$0.320_{\pm 0.000}$	
		720	$0.556_{\pm 0.009}$	$0.442_{\pm 0.006}$	$0.584_{\pm 0.027}$	$0.459_{\pm 0.004}$	$0.396_{\pm 0.012}$	$0.782_{\pm 0.041}$	-	$0.356_{\pm 0.000}$	

Table 3.10: Identity Attention represents our SAMformer with the attention weight matrix constrained to an identity matrix. We report in the last column the **Overall improvement** in MSE and MAE of SAMformer over the Identity Attention. This setup demonstrates that naively fixing the attention matrix to the identity does not enable to match the performance of SAM, despite the near-identity attention matrices SAM showcases. In particular, we observe an overall improvement of 11.93% in MSE and 4.18% in MAE across all the datasets.

Model	Metrics	H	ETTh1	ETTh2	ETTm1	ETTm2	Electricity	Exchange	Traffic	Weather	**Overall Improvement**
Identity Attention	MSE	96	$0.477_{\pm 0.059}$	$0.346_{\pm 0.055}$	$0.345_{\pm 0.027}$	$0.201_{\pm 0.035}$	$0.175_{\pm 0.015}$	$0.179_{\pm 0.031}$	$0.416_{\pm 0.037}$	$0.206_{\pm 0.019}$	11.93%
		192	$0.467_{\pm 0.074}$	$0.374_{\pm 0.031}$	$0.384_{\pm 0.042}$	$0.248_{\pm 0.016}$	$0.189_{\pm 0.022}$	$0.320_{\pm 0.070}$	$0.437_{\pm 0.041}$	$0.236_{\pm 0.002}$	
		336	$0.512_{\pm 0.070}$	$0.372_{\pm 0.024}$	$0.408_{\pm 0.032}$	$0.303_{\pm 0.022}$	$0.211_{\pm 0.019}$	$0.443_{\pm 0.071}$	$0.500_{\pm 0.155}$	$0.277_{\pm 0.003}$	
		720	$0.505_{\pm 0.107}$	$0.405_{\pm 0.012}$	$0.466_{\pm 0.043}$	$0.397_{\pm 0.029}$	$0.233_{\pm 0.019}$	$1.123_{\pm 0.076}$	$0.468_{\pm 0.021}$	$0.338_{\pm 0.009}$	
	MAE	96	$0.473_{\pm 0.041}$	$0.395_{\pm 0.033}$	$0.376_{\pm 0.019}$	$0.294_{\pm 0.027}$	$0.283_{\pm 0.023}$	$0.320_{\pm 0.023}$	$0.301_{\pm 0.039}$	$0.259_{\pm 0.021}$	4.18%
		192	$0.463_{\pm 0.055}$	$0.413_{\pm 0.022}$	$0.399_{\pm 0.030}$	$0.321_{\pm 0.012}$	$0.291_{\pm 0.029}$	$0.418_{\pm 0.043}$	$0.314_{\pm 0.042}$	$0.278_{\pm 0.002}$	
		336	$0.490_{\pm 0.049}$	$0.413_{\pm 0.015}$	$0.411_{\pm 0.019}$	$0.354_{\pm 0.018}$	$0.309_{\pm 0.021}$	$0.498_{\pm 0.041}$	$0.350_{\pm 0.106}$	$0.305_{\pm 0.003}$	
		720	$0.496_{\pm 0.066}$	$0.438_{\pm 0.008}$	$0.444_{\pm 0.030}$	$0.406_{\pm 0.017}$	$0.322_{\pm 0.021}$	$0.788_{\pm 0.021}$	$0.325_{\pm 0.023}$	$0.347_{\pm 0.009}$	

a high versatility and robustness across datasets and prediction horizons. Finally, we also showed that channel-wise attention in time series forecasting can be more efficient – both computationally and performance-wise – than temporal attention commonly used previously. We believe that this surprising finding may spur many further works building on top of our simple architecture to improve it even further.

Chapter 4

On Multi-Task Learning in Multivariate Time Series Forecasting

Summary. We present a novel approach to multivariate time series forecasting by framing it as a multi-task learning problem. We propose an optimization strategy that enhances single-channel predictions by leveraging information across multiple channels. Our framework offers a closed-form solution for linear models and connects forecasting performance to key statistical properties using advanced analytical tools. Empirical results on both synthetic and real-world datasets demonstrate that integrating our method into training loss functions significantly improves univariate models by effectively utilizing multivariate data within a multi-task learning framework.

4.1 Introduction

Multivariate time series forecasting (MTSF) is central to numerous applications involving the simultaneous prediction of multiple interrelated variables, such as medical signals (Čepulionis & Lukoševičiūtė, 2016), energy consumption (UCI, 2015), weather conditions (Max Planck Institute, 2021), and financial markets (Sonkavde et al., 2023). Forecasting accurately in these contexts requires addressing intricate challenges, including capturing cross-channel dependencies, managing long-term temporal correlations, and preventing model overfitting, particularly when data is limited or noisy.

Existing methods for MTSF range from classical statistical models such as ARIMA (George Edward Pelham Box & G. Jenkins, 1990; G. E. P. Box et al., 1974) to sophisticated deep learning architectures, including recurrent and convolutional neural networks (Salinas et al., 2020b; Sen et al., 2019; Guokun Lai et al., 2018c). Recently, Transformer-based architectures have gained popularity due to their success in natural language processing and computer vision (Vaswani et al., 2017; Dosovitskiy et al., 2021b; Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al., 2023b). However,

these architectures encounter significant limitations when directly applied to multivariate time series, often failing to surpass simpler linear approaches (Zeng, M. Chen, et al., 2023; S.-A. Chen et al., 2023). A core limitation arises because many forecasting methods either treat each channel independently or rely heavily on complex decomposition techniques (Yuqi Nie et al., 2023; H. Wu et al., 2021), leading either to suboptimal data utilization or computationally demanding models.

To overcome these limitations, this chapter explicitly frames MTSF as a multi-task learning (MTL) problem (Caruana, 1997), where each channel of a multivariate series corresponds to a distinct but related forecasting task. Under this perspective, cross-channel information is naturally leveraged through shared learning representations, enabling effective utilization of common temporal patterns. Historically, MTL has demonstrated substantial improvements in generalization across various domains, including computer vision (Shao, 2015), natural language processing (Ruder et al., 2019; Raffel et al., 2020), and computational biology (Mei et al., 2011; Shin et al., 2016; Hu et al., 2019). Yet, explicit application of MTL frameworks for multivariate forecasting, particularly for linear and interpretable methods, remains relatively underexplored.

This thesis develops a straightforward yet powerful optimization framework within the MTL paradigm specifically tailored for MTSF. Our method decomposes the forecasting model into shared and task-specific components, explicitly controlling this balance through data-driven hyperparameters. By employing advanced analytical tools, including closed-form solutions for linear settings and precise statistical analyses inspired by Random Matrix Theory (RMT) (Z. Bai & Silverstein, 2010; T. Tao, 2012), we quantify how similarities and differences among channels influence predictive performance. Specifically, we analyze conditions that facilitate positive knowledge transfer—where shared information improves forecasting accuracy—and avoid negative transfer scenarios, where dissimilarities between tasks could degrade performance (F. Yang et al., 2023).

Unlike existing theoretical studies offering broad and often impractical performance bounds (Sai Li et al., 2022; Mousavi Kalan et al., 2020; Nguyen & Couillet, 2023), our methodology delivers concrete guidance for practical hyperparameter tuning directly informed by dataset characteristics. Empirical validations on both synthetic and real-world datasets confirm that our method significantly improves baseline univariate forecasting models such as PatchTST (Yuqi Nie et al., 2023) and DLinear (Zeng, M. Chen, et al., 2023) by effectively capturing inter-channel dependencies. Moreover, our results demonstrate predictive performance comparable to sophisticated multivariate models such as SAMformer (Ilbert et al., 2024) and iTransformer (Yong Liu et al., 2024), all while maintaining simplicity, interpretability, and computational efficiency.

Our Method. We approach MTSF using a multi-task learning framework (Caruana, 1997), where each of the T time series channels is considered as an individual task. For clarity, we denote L as the historical length of the time series, consistent with the previous chapters, and H as the prediction horizon. Additionally, for convenience in this context, we use T to represent the number of tasks. Note that T corresponds to d, the number of chan-

nels, as used in the previous chapter. Each task $t \in \{1, \ldots, T\}$ is defined by an input space $\mathcal{X}^{(t)} \subset \mathbb{R}^L$ and an output space $\mathcal{Y}^{(t)} \subset \mathbb{R}^H$. For each task t, we assume that we are given n_t training examples organized into the feature matrix $\mathbf{X}^{(t)} = [\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_{n_t}^{(t)}] \in \mathbb{R}^{L \times n_t}$ and the corresponding response matrix $\mathbf{Y}^{(t)} = [\mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_{n_t}^{(t)}] \in \mathbb{R}^{H \times n_t}$, where $\mathbf{x}_i^{(t)} \in \mathcal{X}^{(t)}$ represents the i-th feature vector of the t-th task and $\mathbf{y}_i^{(t)} \in \mathcal{Y}^{(t)}$ is the associated response. In particular, we study a straightforward linear signal-plus-noise model that evaluates the response $\mathbf{y}_i^{(t)}$ for the i-th sample of the t-th task as follows:

$$\forall t \in \{1, \ldots, T\}, \quad \mathbf{Y}^{(t)} = \frac{\mathbf{X}^{(t)^\top} \mathbf{W}_t}{\sqrt{T\,d}} + \boldsymbol{\varepsilon}^{(t)} \tag{4.1}$$

where $\boldsymbol{\varepsilon}^{(t)} \in \mathbb{R}^{n_t \times H}$ is a matrix of noise vectors with each $\boldsymbol{\varepsilon}_i^{(t)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_N)$, $\boldsymbol{\Sigma}_N \in \mathbb{R}^{H \times H}$ denoting the covariance matrix.

The matrix $\mathbf{W}_t \in \mathbb{R}^{L \times H}$ denotes the signal-generating hyperplane for task t. We denote the concatenation of all task-specific hyperplanes by $\mathbf{W} = [\mathbf{W}_1^\top, \ldots, \mathbf{W}_T^\top]^\top \in \mathbb{R}^{TL \times H}$. We assume that $\mathbf{W}_t$ can be decomposed as a sum of a common matrix $\mathbf{W}_0 \in \mathbb{R}^{L \times H}$, which captures the shared information across all the tasks, and a task-specific matrix $\mathbf{V}_t \in \mathbb{R}^{L \times H}$, which captures deviations specific to the task t:

$$\mathbf{W}_t = \mathbf{W}_0 + \mathbf{V}_t. \tag{4.2}$$

Given the multitask regression framework and the linear signal-plus-noise model, we now want to retrieve the common and specific hyperplanes, $\mathbf{W}_0$ and $\mathbf{V}_t$, respectively. To achieve this, we study the following minimization problem governed by a parameter λ that controls the balance between the common and specific components of $\mathbf{W}$:

$$(\mathbf{W}_0^*, \{\mathbf{V}_t^*\}_{t=1}^T, \lambda^*) = \arg \min_{\mathrm{W}_0, \{\mathrm{V}_t\}, \lambda} \left[\frac{1}{2\lambda} \|\mathbf{W}_0\|_F^2 + \frac{1}{2} \sum_{t=1}^T \frac{\|\mathbf{V}_t\|_F^2}{\gamma_t} + \frac{1}{2} \sum_{t=1}^T \left\| \mathbf{Y}^{(t)} - \frac{\mathbf{X}^{(t)^\top} \mathbf{W}_t}{\sqrt{TL}} \right\|_F^2 \right] \tag{4.3}$$

where $\gamma = [\gamma_1, \ldots, \gamma_T]$ is a vector of task-specific regularization hyperparameters. Each γ_t controls how much the model overfits (small γ_t) or underfits (large γ_t) on task t.

Contributions. Our contributions are as follows:

1. We formalize MTSF as an MTL problem, providing an optimization framework with closed-form solutions in linear contexts.

2. We propose a practical, data-driven method for selecting hyperparameters that control the balance between shared and task-specific components, directly informed by data statistics.

3. Through extensive empirical evaluation, we demonstrate our framework's superiority over existing single-task baselines and competitiveness with state-of-the-art multivariate models.

4. Our approach improves robustness and interpretability by explicitly leveraging cross-channel similarities, which traditional forecasting methods neglect.

Results. We validate our framework on multivariate time series forecasting, showing that our multi-task learning approach enhances univariate forecasting models like PatchTST (Yuqi Nie et al., 2023) and DLinear (Zeng, M. Chen, et al., 2023) by leveraging shared learning across channels. Our method achieves performance comparable to state-of-the-art multivariate models such as SAMformer (Ilbert et al., 2024) and iTransformer (Yong Liu et al., 2024), demonstrating the effectiveness of treating MTSF as a multi-task learning problem.

4.2 Framework

Notations. Throughout this study, matrices are represented by bold uppercase letters (e.g., matrix $\mathbf{A}$), vectors by bold lowercase letters (e.g., vector $\mathbf{v}$), and scalars by regular, non-bold typeface (e.g., scalar a). The notation $\mathbf{A} \otimes \mathbf{B}$ for matrices or vectors $\mathbf{A}, \mathbf{B}$ is the Kronecker product. $\mathcal{D}_x$ or $\mathrm{Diag}(\mathbf{x})$ stands for a diagonal matrix containing on its diagonal the elements of the vector $\mathbf{x}$. The superscripts t and i are used to denote the task and the sample number, respectively, e.g., $\mathbf{x}_i^{(t)}$ writes the i-th sample of the t-th task. The canonical vector of $\mathbb{R}^T$ is denoted by $\mathbf{e}_t^{[T]}$ with $[e_t^{[T]}]_i = \delta_{ti}$. Given a matrix $M \in \mathbb{R}^{p \times n}$, the Frobenius norm of $\mathbf{M}$ is denoted $\|\mathbf{M}\|_F \equiv \sqrt{tr(\mathbf{M}^\top \mathbf{M})}$. For our theoretical analysis, we introduce the following notation of training data:

$$\mathbf{Y} = [\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(T)}] \in \mathbb{R}^{H \times n}, \qquad \mathbf{Z} = \sum_{t=1}^{T} \left(\mathbf{e}_t^{[T]} \mathbf{e}_t^{[T]^\top} \right) \otimes \mathbf{X}^{(t)} \in \mathbb{R}^{TL \times n}$$

where $n = \sum_{t=1}^{T} n_t$ is the total number of samples in all the tasks.

4.2.1 Multi-Task Model

We solve the multi-task forecasting problem by finding $\hat{\mathbf{W}} = [\hat{\mathbf{W}}_1^\top, \ldots, \hat{\mathbf{W}}_T^\top]^\top \in \mathbb{R}^{LT \times H}$ under the assumption $\mathbf{W}_t = \mathbf{W}_0 + \mathbf{V}_t$, where $\mathbf{W}_0$ is the shared component. The optimization problem is:

$$\min_{(\mathrm{W}_0, \mathrm{V}) \in \mathbb{R}^{L \times H} \times \mathbb{R}^{LT \times H}} \mathcal{J}(\mathbf{W}_0, \mathbf{V}),$$

where

$$\mathcal{J}(\mathbf{W}_0, \mathbf{V}) = \frac{1}{2\lambda} \|\mathbf{W}_0\|_F^2 + \frac{1}{2} \sum_{t=1}^{T} \frac{\|\mathbf{V}_t\|_F^2}{\gamma_t} + \frac{1}{2} \sum_{t=1}^{T} \left\| \mathbf{Y}^{(t)} - \frac{\mathbf{X}^{(t)^\top} \mathbf{W}_t}{\sqrt{TL}} \right\|_F^2.$$

This convex optimization problem has a unique solution as shown in Appendix B.1.

4.2.2 Assumptions

In order to use Random Matrix Theory (RMT) tools, as shown in Appendix B, we make two assumptions on the data distribution and the asymptotic regime. Following Wainwright 2019, we adopt a concentration hypothesis on the feature vectors $\mathbf{x}_i^{(t)}$, which was shown to be highly effective for analyzing machine learning problems (Couillet & Liao, 2022; Feofanov, Tiomoko, et al., 2023).

Assumption 4.2.1 (Concentrated Random Vector). *We assume that there exist two constants $C, c > 0$ (independent of dimension d) such that, for any task t, for any 1-Lipschitz function $f : \mathbb{R}^L \to \mathbb{R}$, any feature vector $\mathbf{x}^{(t)} \in \mathcal{X}^{(t)}$ verifies:*

$$\forall t > 0 : \mathbb{P}(|f(\mathbf{x}^{(t)}) - \mathbb{E}[f(\mathbf{x}^{(t)})]| \geq t) \leq Ce^{-(t/c)^2},$$
$$\mathbb{E}[\mathbf{x}^{(t)}] = 0 \quad \textit{and} \quad \mathrm{Cov}[\mathbf{x}^{(t)}] = \mathbf{\Sigma}^{(t)}.$$

In particular, we distinguish the following scenarios: $\mathbf{x}_i^{(t)} \in \mathbb{R}^L$ are concentrated when they are (i) independent Gaussian random vectors with covariance of bounded norm, (ii) independent random vectors uniformly distributed on the $\mathbb{R}^L$ sphere of radius $\sqrt{L}$, and most importantly (iii) any Lipschitz transformation $\phi(\mathbf{x}_i^{(t)})$ of the above two cases, with bounded Lipschitz norm. Scenario (iii) is especially pertinent for modeling data in realistic settings. Recent research (Seddik et al., 2020) has demonstrated that images produced by generative adversarial networks (GANs) are inherently qualified as concentrated random vectors.

Next, we present a classical RMT assumption that establishes a commensurable relationship between the number of samples and dimension.

Assumption 4.2.2 (High-dimensional asymptotics). *As $L \to \infty$, $n_t = \mathcal{O}(L)$ and $T = \mathcal{O}(1)$. More specifically, we assume that $n/L \xrightarrow{a.s.} c_0 < \infty$ with $n = \sum_{t=1}^{T} n_t$.*

Although different from classical asymptotic where the number of samples is implicitly assumed to be exponentially larger than the dimension, the high-dimensional asymptotic finds many applications including telecommunications (Couillet & Debbah, 2011), finance (Potters et al., 2005) and machine learning (Couillet & Liao, 2022; Tiomoko et al., 2020; Feofanov, Tiomoko, et al., 2023).

4.2.3 Discussions on the Assumptions.

On the zero-mean assumption. We would like to note that we are assuming that both the noise $\boldsymbol{\varepsilon}$ and the feature $\mathbf{x}_i^{(t)}$ have zero mean. This is a common assumption in many statistical models and it simplifies the analysis. However, this assumption is not restrictive. In practice, if the data or the response variable are not centered, we can always preprocess the data by subtracting the mean. This preprocessing step brings us back to the zero-mean setting that we consider in our theoretical analysis.

On the Assumption 1. Data are concentrated random vectors, meaning high-dimensional data maintain stable under complex (Lipschitz) transformations. The strong performance of neural networks on tasks like image recognition and NLP suggests that these models produce stable predictions. As Lipschitz transformations, they maintain controlled distances between inputs, ensuring stability. Recent studies have demonstrated and experimentally confirmed that both real-world data and synthetically generated data using GANs exhibit concentration properties, supporting this assumption. This makes our assumption more realistic than traditional Gaussian assumptions, as it does not rely on specific hypotheses about the shape of the data distribution, but rather on the stability of statistical properties after transformation. Consequently, analyzing a framework of concentrated random vectors is more theoretically challenging than using Gaussian assumptions and represents a key novelty of our theory.

On the Assumption 2. The dimension L is of the same order of magnitude as the sample size n. This joint growth captures data complexity better than assuming a fixed feature size with increasing samples, which can oversimplify models. Our theory works for fixed L and n, unbiased by specific parameter choices. The accuracy of empirical predictions depends on both L and n, with variance scaling as $\mathcal{O}\left(\frac{1}{\sqrt{nL}}\right)$. Larger L and n reduce variance, making empirical results more reliable and closer to theoretical values. Conversely, smaller L and n increase variance, affecting single predictions. However, this scaling is still better than n growing indefinitely with fixed L, where variance scales as $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$, leading to increased bias.

4.3 Main Theoretical Results

4.3.1 Estimation of the Performances

Given training data $\mathbf{X} \in \mathbb{R}^{n \times L}$ and response $\mathbf{Y} \in \mathbb{R}^{n \times H}$, we define the training and test risks as:

$$\mathcal{R}^{\infty}_{train} = \frac{1}{Tn}\mathbb{E}\left[\|\mathbf{Y} - g(\mathbf{X})\|_2^2\right], \quad \mathcal{R}^{\infty}_{test} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\mathbf{y}^{(t)} - g(\mathbf{x}^{(t)})\|_2^2\right],$$

with

$$g(\mathbf{x}^{(t)}) = \frac{1}{TL}\left(\mathbf{e}_t^{[T]} \otimes \mathbf{x}^{(t)}\right)^{\top} \mathbf{AZQY},$$

where we set:

$$\mathbf{Q} = \left(\frac{\mathbf{Z}^{\top}\mathbf{AZ}}{TL} + \mathbf{I}_{TL}\right)^{-1}, \quad \mathbf{A} = \left(\mathcal{D}_{\boldsymbol{\gamma}} + \lambda \mathbb{1}_T \mathbb{1}_T^{\top}\right) \otimes \mathbf{I}_L \in \mathbb{R}^{TL \times TL}.$$

To analyze $\mathcal{R}^{\infty}_{train}$ and $\mathcal{R}^{\infty}_{test}$, we use a deterministic equivalent of $\mathbf{Q}$, denoted $\bar{\mathbf{M}}$, which approximates $\mathbf{Q}$ in a linear form. This approach allows estimating key quantities like $\frac{1}{L}\mathrm{tr}(\mathbf{AQ})$ using the coresolvent $\tilde{\mathbf{Q}} = \left(\frac{\mathbf{A}^{\frac{1}{2}}\mathbf{Z}\mathbf{Z}^{\top}\mathbf{A}^{\frac{1}{2}}}{TL} + \mathbf{I}_{TL}\right)^{-1}$.

Using Lemma 1 provided in the Appendix B.2.1, whose proofs are included in Appendices B.2.2, B.2.3 and B.2.4, we establish the deterministic equivalents that allow us to introduce our Theorem 4.3.1, characterizing the asymptotic behavior of both training and testing risks.

Theorem 4.3.1 (Asymptotic test risk). *Under the assumptions of concentrated random vectors and high-dimensional asymptotics, the asymptotic test risk is given by:*

$$\mathcal{R}^{\infty}_{test} = \underbrace{\frac{\mathrm{tr}\left(\mathbf{W}^{\top}\mathbf{A}^{-\frac{1}{2}}\,\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A})\,\mathbf{A}^{-\frac{1}{2}}\mathbf{W}\right)}{TL}}_{\textit{signal term}} + \underbrace{\frac{\mathrm{tr}\left(\boldsymbol{\Sigma}_n\,\bar{\mathbf{Q}}_2\right)}{TL} + \mathrm{tr}\left(\boldsymbol{\Sigma}_n\right)}_{\textit{noise terms}} \tag{ATR}$$

where $\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A})$ and $\bar{\mathbf{Q}}_2$ are deterministic equivalents for specific matrix forms of $\tilde{\mathbf{Q}}$ and $\mathbf{Q}$.

We defer the full proof of this theorem to Appendix B.3.

4.3.2 Error Contribution Analysis

To gain theoretical insights, we analyze (ATR) consisting of the signal and the noise components.

Signal Term. The signal term can be further approximated, up to some constants as $tr(\mathbf{W}^\top(\mathbf{A\Sigma}+\mathbf{I})^{-2}\mathbf{W})$ with $\mathbf{\Sigma} = \sum_{t=1}^{T} \frac{n_t}{L}\mathbf{\Sigma}^{(t)}$. The matrix $(\mathbf{A\Sigma}+\mathbf{I})^{-2}$ plays a crucial role in amplifying the signal term $tr(\mathbf{W}^\top\mathbf{W})$, which in turn allows the test risk to decrease. The off-diagonal elements of $(\mathbf{A\Sigma}+\mathbf{I})^{-2}$ amplify the cross terms ($tr(\mathbf{W}_v^\top\mathbf{W}_t)$ for $t \neq v$), enhancing the multi-task aspect, while the diagonal elements amplify the independent terms ($\|\mathbf{W}_t\|_2^2$). This structure is significant in determining the effectiveness of multi-task learning. Furthermore, both terms decrease with an increasing number of samples n_t, smaller values of γ_t, and a larger value of λ. The cross term, which is crucial for multi-task learning, depends on the matrix $\mathbf{\Sigma}_t^{-1}\mathbf{\Sigma}_v$. This matrix represents the shift in covariates between tasks. When the features are aligned (i.e., $\mathbf{\Sigma}_t^{-1}\mathbf{\Sigma}_v = \mathbf{I}_L$), the cross term is maximized, enhancing multi-task learning. However, a larger Fisher distance between the covariances of the tasks results in less favorable correlations for multi-task learning.

Noise term. Similar to the signal term, the noise term can be approximated, up to some constants, as $tr(\mathbf{\Sigma}_N(\mathbf{A}^{-1}+\mathbf{\Sigma})^{-1})$. However, there is a major difference between the way both terms are expressed in the test risk. The noise term does not include any cross terms because the noise across different tasks is independent. In this context, only the diagonal elements of the matrix are significant. This diagonal term increases with the sample size and the value of λ. It is responsible for what is known as negative transfer. As the diagonal term increases, it negatively affects the transfer of learning from one task to another. This is a critical aspect to consider in multi-task learning scenarios.

4.3.3 Simplified Model for Clear Insights

In this section, we specialize the theoretical analysis to the simple case of two tasks ($T = 2$) on the *Appliance Energy* dataset. The results are presented in Figure 4.1, confirming that the test risk follows a convex curve, allowing the identification of an optimal value λ^*. First, we assume that the tasks share the same identity covariance and that $\gamma_1 = \gamma_2 \equiv \gamma$. Under these conditions, the test risk can be approximated, up to some constants, as

$$\mathcal{R}_{test}^{\infty} = \mathbf{D}_{IL}\left(\|\mathbf{W}_1\|_2^2 + \|\mathbf{W}_2\|_2^2\right) + \mathbf{C}_{MTL}\mathbf{W}_1^\top\mathbf{W}_2 + \mathbf{N}_{NT}\mathrm{tr}(\mathbf{\Sigma}_n)$$

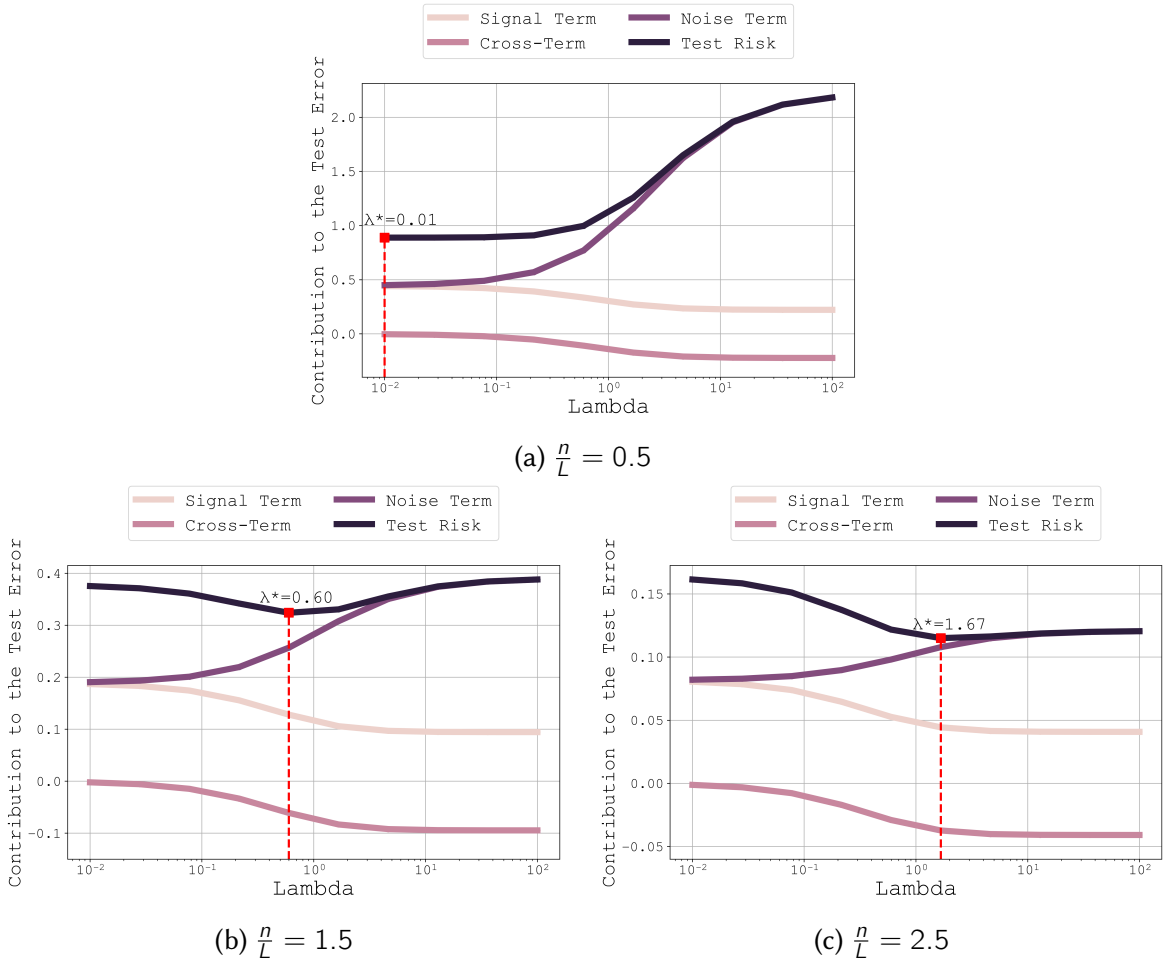
where the diagonal term (independent learning) $\mathbf{D}_{IL}$, the cross term (multi-task learning) $\mathbf{C}_{MTL}$, and the noise term (negative transfer) $\mathbf{N}_{NT}$ have closed-form expressions depending on γ and λ:

$$\mathbf{D}_{IL} = \frac{(c_0(\lambda+\gamma)+1)^2 + c_0^2\lambda^2}{(c_0(\lambda+\gamma)+1)^2 - c_0^2\lambda^2}, \quad \mathbf{C}_{MTL} = \frac{-2c_0\lambda(c_0(\lambda+\gamma)+1)}{(c_0(\lambda+\gamma)+1)^2 - c_0^2\lambda^2}$$
$$\mathbf{N}_{NT} = \frac{(c_0(\lambda+\gamma)^2 + (\lambda+\gamma) - c_0\lambda^2)^2 + \lambda^2}{((c_0(\lambda+\gamma)+1)^2 - c_0^2\lambda^2)^2}$$

We recall that c_0 has been defined in the Assumption 4.2.2. As previously mentioned, the test risk is primarily composed of two terms: the signal term and the noise term, which are in competition with each other. The more similar the tasks are, the stronger the signal term becomes. In the following plot, we illustrate how this competition can influence

the risk. Depending on the value of the parameter λ and the sample sizes, the risk can either decrease monotonically, increase monotonically, or exhibit a convex behavior. This competition can lead to an optimal value for λ, which interestingly has a simple closed-form expression that can be obtained by deriving the $\mathcal{R}^{\infty}_{test}$ w.r.t. λ as follows (see details in Appendix B.4.4):

$$\lambda^{\star} = \frac{n}{L} SNR - \frac{\gamma}{2}, \text{ with } SNR = \frac{\|\mathbf{W}_1\|_2^2 + \|\mathbf{W}_2\|_2^2}{\mathrm{tr}(\mathbf{\Sigma}_n)} + \frac{\mathbf{W}_1^{\top}\mathbf{W}_2}{\mathrm{tr}(\mathbf{\Sigma}_n)}.$$



(a) $\frac{n}{L} = 0.5$

(b) $\frac{n}{L} = 1.5$

(c) $\frac{n}{L} = 2.5$

Figure 4.1: Test loss contributions $\mathbf{D}_{IL}$, $\mathbf{C}_{MTL}$, $\mathbf{N}_{NT}$ across three sample size regimes. Test risk exhibits decreasing, increasing, or convex shapes based on the regime. λ^* from theory are marked.

4.3.4 Comparison between Empirical and Theoretical Predictions

In this section, we compare the theoretical predictions with the empirical results on synthetic data. Our experiment is based on a two-task setting ($T = 2$) defined as $\mathbf{W}_1 \sim$

$\mathcal{N}(0, I_p)$ with $\mathbf{W}_2 = \alpha\mathbf{W}_1 + \sqrt{1-\alpha^2}\mathbf{W}_1^{\perp}$. $\mathbf{W}_1^{\perp}$ represents any vector orthogonal to $\mathbf{W}_1$ and $\alpha \in [0, 1]$. This setting allows us to adjust the similarity between tasks through α.

Figure 4.2 shows a comparison of the theoretical and empirical classification errors for different values of λ, highlighting the error-minimizing value of λ. Despite the relatively small values of n and p, there is a very precise match between the asymptotic theory and the practical experiment. This is particularly evident in the accurate estimation of the optimal value for λ.

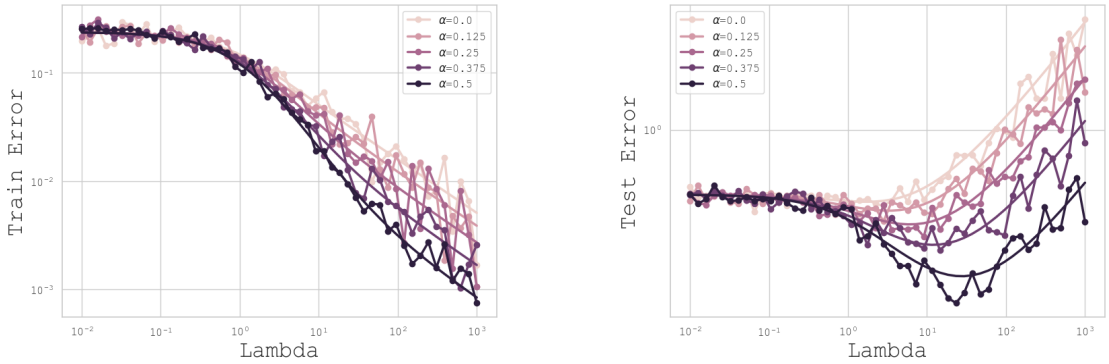


Figure 4.2: Empirical and theoretical train and test MSE as functions of the parameter λ for different values of α. The smooth curves represent the theoretical predictions, while the corresponding curves with the same color show the empirical results, highlighting that the empirical observations indeed match the theoretical predictions.

4.4 Experimental Results

4.4.1 Relevance of the theoretical insights beyond the case of linear models

While non-linear models are widely used, establishing their theoretical foundations is challenging. Therefore, we focused on linear models, which, despite their simplicity, provide valuable insights into more complex models.

Our results show that test risk curves for non-linear models follow patterns predicted by our theory. This is expected because non-linear models in time series forecasting typically use a linear output layer for prediction. Thus, we can apply our theory to the inputs of this final linear layer. This approach is valid due to data concentration and the Lipschitz nature of neural networks, ensuring outputs of the non-linear part don't deviate significantly from the inputs.

Moreover, multivariate time series models often treat channels separately using univariate methods, missing cross-channel information. Our results in Section 4.4.2 show

that our method surpasses univariate baselines by optimally regularizing with λ and γ, supporting our theory's applicability to non-linear models as the final linear layer effectively leverages concentrated inputs.

Finally, our regularization approach differs from traditional cross-task regularizations that use one task per dataset. We consider each prediction as a task and introduce γ_t parameters alongside λ. These parameters enforce multivariate regularization and control underfitting or overfitting per task. This method is tractable since it's applied at the model's final layer.

The similarity between curves for non-linear and linear models indicates our findings are robust; non-linear models also exhibit optimal regularization parameters, enhancing performance in multivariate forecasting.

4.4.2 Application to Multivariate Time Series Forecasting

Our theoretical framework is applied in the context of Multivariate Time Series Forecasting. We previously applied this framework in a linear setting, and now aim to evaluate its empirical validity in the non-linear setting of neural networks. The results presented in this section represent the best test MSE, assuming the ability to find the optimal lambda value, which can be considered as an oracle scenario. A study of these limitations can be found in Appendix B.8.

Motivation. Our approach is applied to the MTSF setting for several reasons. Firstly, many models currently used are essentially univariate, where predictions for individual series are simply concatenated without exploiting the multivariate information inherent in traditional benchmarks. Given that these benchmarks are designed for multivariate forecasting, leveraging multivariate information should yield better results. Secondly, our theoretical framework can benefit this domain, as most predictive models use a linear layer on top of the model to project historical data of length L for predicting the output of length H. This characteristic aligns well with our method, making it a promising fit for enhancing forecasting accuracy.

Our approach. We propose a novel method for MTSF by modifying the loss function to incorporate both individual feature transformations f_t and a shared transformation f_0. Each univariate-specific transformation f_t is designed to capture the unique dynamics of its respective feature, while f_0 serves as a common transformation applied across all features to capture underlying patterns shared among them. We consider a neural network f with inputs $\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(T)}]$, where T is the number of channels and $\mathbf{X}^{(t)} \in \mathbb{R}^{n \times L}$. For a univariate model without MTL regularization, we predict $\mathbf{Y} = [\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots, \mathbf{Y}^{(T)}] = [f_1(\mathbf{X}^{(1)}), \dots, f_T(\mathbf{X}^{(T)})]$ and $\mathbf{Y}^{(t)} \in \mathbb{R}^{n \times H}$. We compare these models with their corresponding versions that include MTL regularization, formulated as: $f_t^{MTL}(\mathbf{X}^{(t)}) = f_t(\mathbf{X}^{(t)}) + f_0(\mathbf{Y})$ with $f_t : \mathbb{R}^{n \times L} \to \mathbb{R}^{n \times H}$ and $f_0 : \mathbb{R}^{n \times HT} \to \mathbb{R}^{n \times H}$. We

define our regularized loss as follows:

$$\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \sum_{t=1}^{T} \|\mathbf{Y}^{(t)} - f_t^{MTL}(\mathbf{X}^{(t)})\|_F^2 + \lambda \|f_0(\mathbf{X})\|_F^2 + \sum_{t=1}^{T} \gamma_t \|f_t(\mathbf{X}^{(t)}\|_F^2, \quad \forall t \in \{1, \dots, T\}.$$

where $\mathbf{Y}^{(t)}$ are the true predictions, f_t represents the univariate model for each channel t, and λ is our regularization parameter, for which we have established a closed form in the case of linear f_t. f_0 serves a role equivalent to W_0, which was defined in our theoretical study and allows for the regularization of the common part. This component can be added at the top of a univariate model. The parameters γ_t enable the regularization of the specialized parts f_t.

In our setup, f_t is computed in a similar way as in the model without regularization and f_0 is computed by first flattening the concatenation of the predictions of $\mathbf{X}^{(t)}$, then applying a linear projection leveraging common multivariate information before reshaping. The loss function is specifically designed to balance fitting the multivariate series using f_0 and the specific channels using f_t. This approach enhances the model's generalization across various forecasting horizons and datasets.

Architectures with MTL Regularization. We implemented the univariate PatchTST, DLinearU, and Transformer baselines with MTL regularization. Initially, we scale the inputs twice using RevIN normalization. The first scaling is applied to the univariate components, and the second scaling is applied to the multivariate components. For each channel, we then apply our model without MTL regularization. The outputs are concatenated along the channel dimension, and this concatenation is flattened to form a matrix of shape (batch size, $H \times T$), where H is the prediction horizon and T is the number of channels. We then learn a square matrix W of shape $(H \times T) \times (H \times T)$ for projection and reshape the result to obtain an output of shape (batch size, H, T). This method can be applied on top of any univariate model.

Datasets. We conduct our experiments on 3 publicly available datasets of real-world time series, widely used for multivariate long-term forecasting (H. Wu et al., 2021; S.-A. Chen et al., 2023; Yuqi Nie et al., 2023). The 2 Electricity Transformer Temperature datasets ETTh1, and ETTh2 (H. Zhou et al., 2021) contain the time series collected by electricity transformers from July 2016 to July 2018. Whenever possible, we refer to this set of 2 datasets as ETT. Weather (Max Planck Institute, 2021) contains the time series of meteorological information recorded by 21 weather indicators in 2020. It should be noted Weather is large-scale datasets. The ETT datasets can be downloaded here while the Weather dataset can be downloaded here. Table 4.1 summarizes the characteristics of the datasets used in our experiments. The results of our 3 baselines on ETTh1 can be found in Appendix B.7.

Training parameters. The training/validation/test split is 12/4/4 months on the ETT datasets and 70%/20%/10% on the Weather dataset. We use a look-back window $L =$

Table 4.1: Characteristics of the multivariate time series datasets used in our experiments.

Dataset	ETTh1/ETTh2	Weather
# features	7	21
# time steps	17420	52696
Granularity	1 hour	10 minutes

336 for PatchTST and $L = 512$ for DLinearU and Transformer, using a sliding window with stride 1 to create the sequences. The training loss is the MSE. Training is performed during 100 epochs and we use early stopping with a patience of 5 epochs. For each dataset, baselines, and prediction horizon $H \in \{96, 192, 336, 720\}$, each experiment is run 3 times with different seeds, and we display the average of the test MSE over the 3 trials in Table 4.3.

Table 4.2: Learning rates used in our experiments.

Dataset	ETTh1/ETTh2	Weather
Learning rate	0.001	0.0001

Results. We present experimental results on different forecasting horizons, using 3 common benchmark MTSF datasets. Our models include PatchTST (Yuqi Nie et al., 2023), known to be on par with state-of-the-art in MTSF while being a univariate model, a univariate DLinear version called DLinearU compared to its multivariate counterpart DLinearM (Zeng, M. Chen, et al., 2023), and a univariate Transformer (Ilbert et al., 2024) with temporal-wise attention compared to the multivariate state-of-the-art models SAMformer (Ilbert et al., 2024) and iTransformer (Yong Liu et al., 2024). Table 4.3 provides a detailed comparison of the test mean squared errors (MSE) for different MTSF models, emphasizing the impact of MTL regularization. Models with MTL regularization are compared to their versions without regularization, as well as SAMformer and iTransformer.

Adding MTL regularization improves the performance of PatchTST, DLinearU, and Transformer in most cases. When compared to state-of-the-art multivariate models, the MTL-regularized models are often competitive. SAMformer is outperformed by at least one MTL-regularized method per horizon and dataset, except for ETTh1 with horizons of 336 and 720. iTransformer is consistently outperformed by at least one MTL-regularized method regardless of the dataset and horizon.

The best performing methods are PatchTST and DLinearU with MTL regularization. These models not only outperform their non-regularized counterparts, often significantly, as shown by Student's t-tests with a p-value of 0.05, but also surpass state-of-the-art multivariate models like SAMformer and Transformer. This superior performance is indicated by the bold values in the table.

Table 4.3: MTL regularization results. Algorithms marked with † are state-of-the-art multivariate models and serve as baseline comparisons. All others are univariate. We compared the models with MTL regularization to their corresponding versions without regularization. Each MSE value is derived from 3 different random seeds. MSE values marked with * indicate that the model with MTL regularization performed significantly better than its version without regularization, according to a Student's t-test with a p-value of 0.05. MSE values are in **bold** when they are the best in their row, indicating the top-performing models.

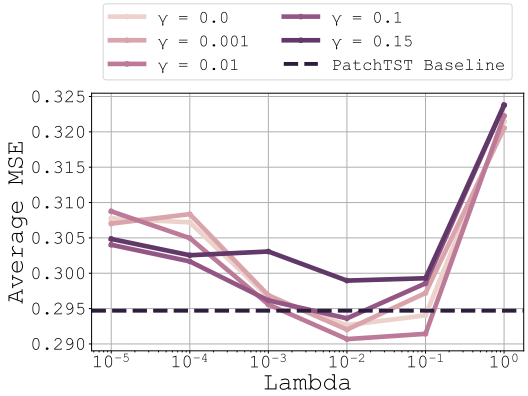
Dataset	H	with MTL regularization			without MTL regularization					
		PatchTST	DLinearU	Transformer	PatchTST	DLinearU	DLinearM	Transformer	SAMformer†	iTransformer†
ETTh1	96	0.385	**0.367***	0.368	0.387	0.397	0.386	0.370	0.381	0.386
	192	0.422	**0.405***	0.407*	0.424	0.422	0.437	0.411	0.409	0.441
	336	0.433*	0.431	0.433	0.442	0.431	0.481	0.437	**0.423**	0.487
	720	0.430*	0.454	0.455*	0.451	0.428	0.519	0.470	**0.427**	0.503
ETTh2	96	0.291	**0.267***	0.270	0.295	0.294	0.333	0.273	0.295	0.297
	192	0.346*	**0.331***	0.337	0.351	0.361	0.477	0.339	0.340	0.380
	336	**0.332***	0.367	0.366*	0.342	0.361	0.594	0.369	0.350	0.428
	720	**0.384***	0.412	0.405*	0.393	0.395	0.831	0.428	0.391	0.427
Weather	96	**0.148**	0.149*	0.154*	0.149	0.196	0.196	0.170	0.197	0.174
	192	**0.190**	0.206*	0.198*	0.193	0.243	0.237	0.214	0.235	0.221
	336	**0.242***	0.249*	0.258	0.246	0.283	0.283	0.260	0.276	0.278
	720	**0.316***	0.326*	0.331	0.322	0.339	0.345	0.326	0.334	0.358

Finally, MTL regularization enhances the performance of univariate models, making them often competitive with state-of-the-art multivariate methods like SAMformer and iTransformer. This approach seems to better capture shared dynamics among tasks, leading to more accurate forecasts.

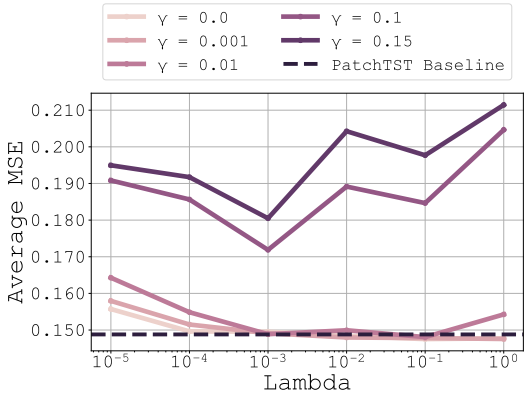
4.5 Conclusions and Future Work

In this chapter, we formulated the problem of multivariate time series forecasting from the perspective of multi-task learning. This innovative approach relies on an optimization strategy that enhances individual channel predictions by effectively leveraging shared information among them. We developed a closed-form analytical solution for linear models and presented a thorough statistical analysis based on random matrix theory. This analysis highlighted the conditions under which a positive transfer of information between tasks can be achieved while avoiding negative transfer scenarios.

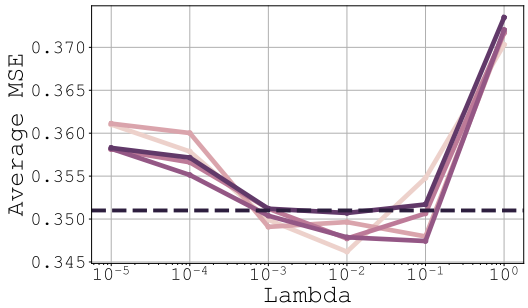
Our theoretical results provide a precise interpretation of the mechanisms involved in multi-task learning applied to time series, clearly establishing the role of regularization parameters in balancing shared and task-specific contributions. In particular, we demon-



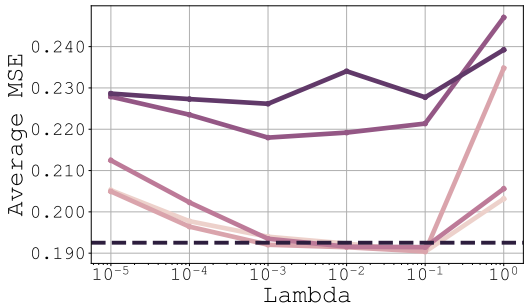
(a) ETTh2 - Horizon 96



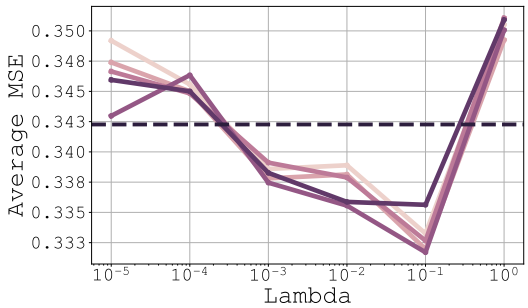
(b) Weather - Horizon 96



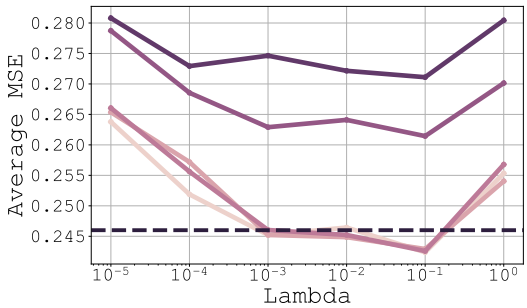
(c) ETTh2 - Horizon 192



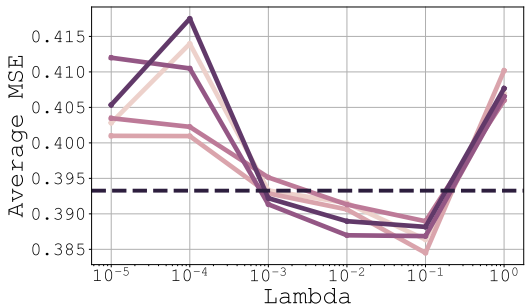
(d) Weather - Horizon 192



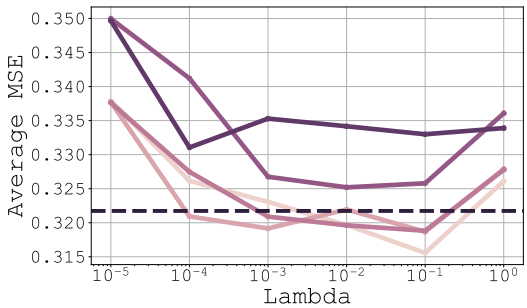
(e) ETTh2 - Horizon 336



(f) Weather - Horizon 336

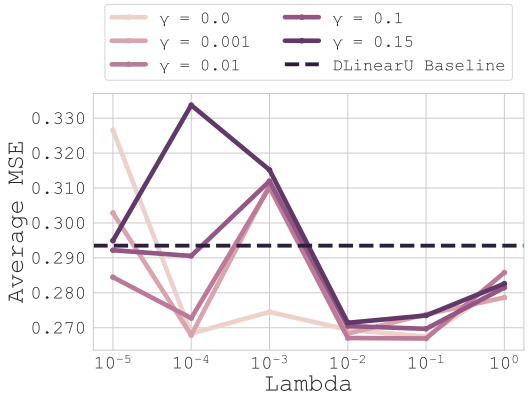


(g) ETTh2 - Horizon 720

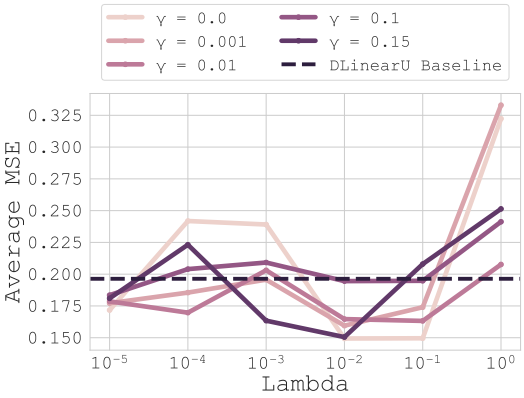


(h) Weather - Horizon 720

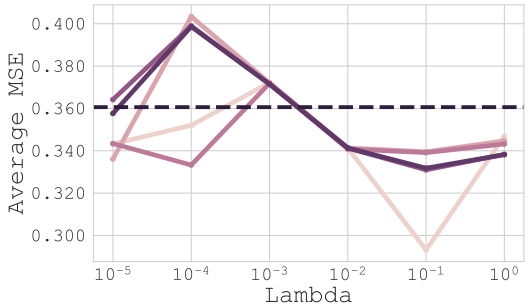
Figure 4.3: Results for datasets ETTh2 and Weather on the PatchTST baseline, averaged across 3 seeds for each gamma and lambda setting.



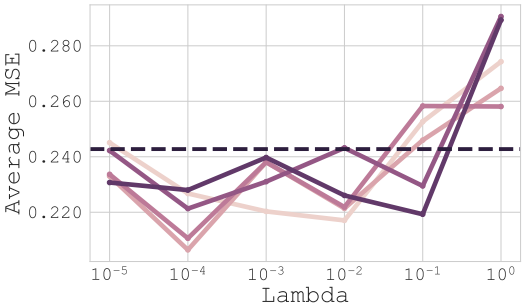
(a) ETTh2 - Horizon 96



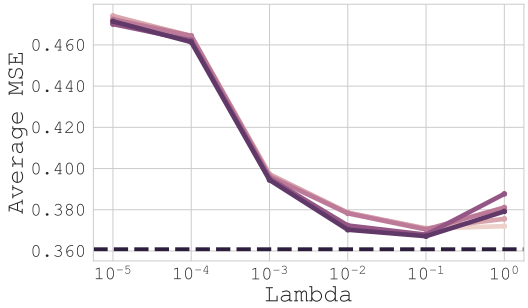
(b) Weather - Horizon 96



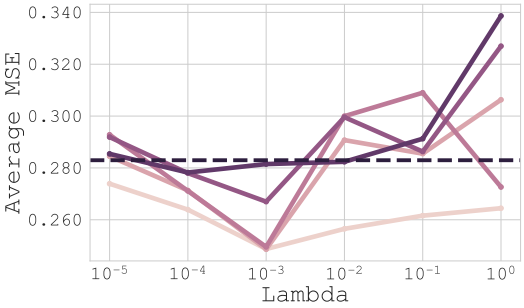
(c) ETTh2 - Horizon 192



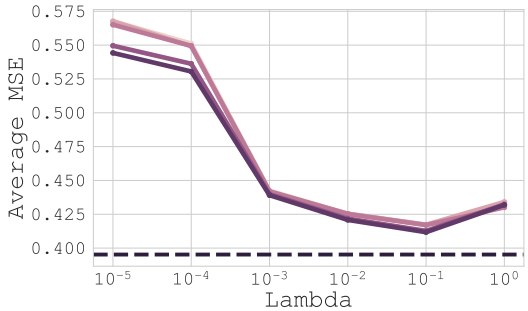
(d) Weather - Horizon 192



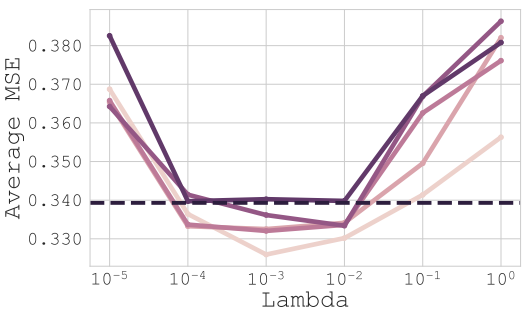
(e) ETTh2 - Horizon 336



(f) Weather - Horizon 336

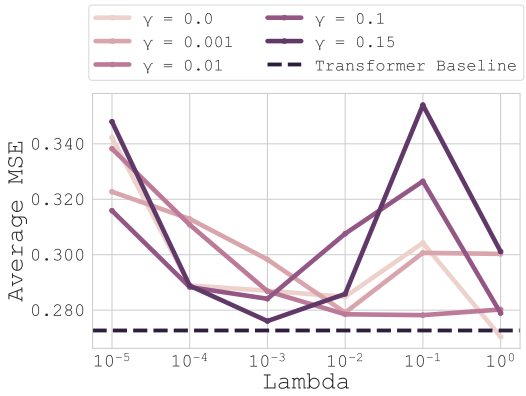


(g) ETTh2 - Horizon 720

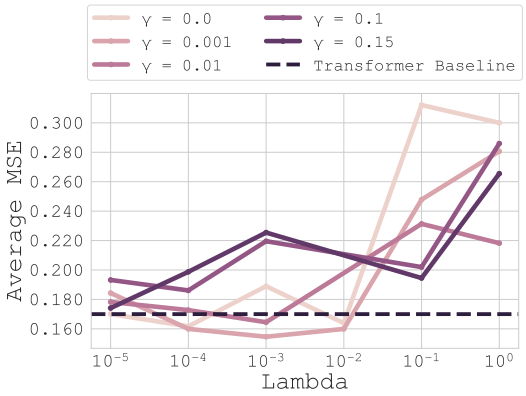


(h) Weather - Horizon 720

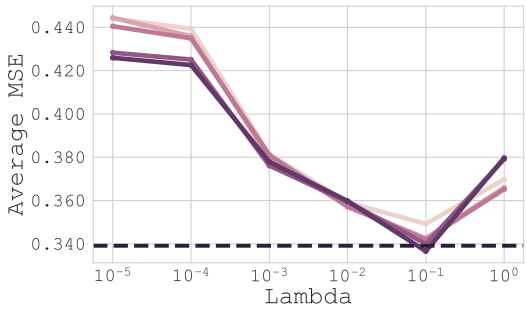
Figure 4.4: Results for datasets ETTh2 and Weather on the DLinearU baseline., averaged across 3 seeds for each gamma and lambda setting.



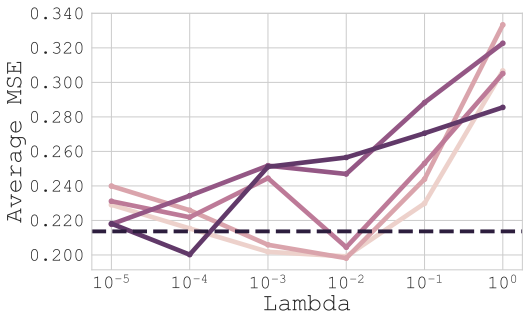
(a) ETTh2 - Horizon 96



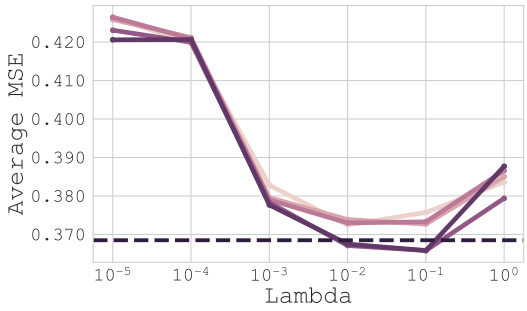
(b) Weather - Horizon 96



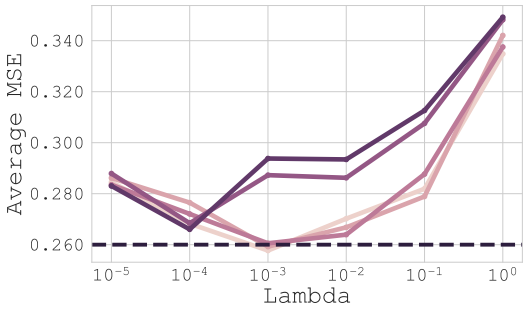
(c) ETTh2 - Horizon 192



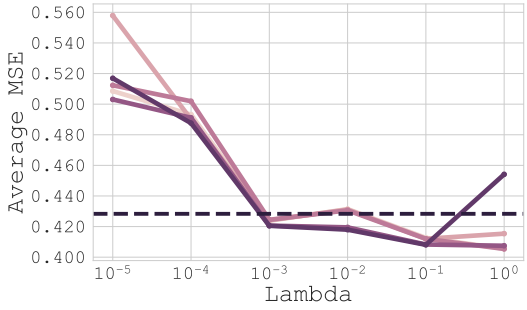
(d) Weather - Horizon 192



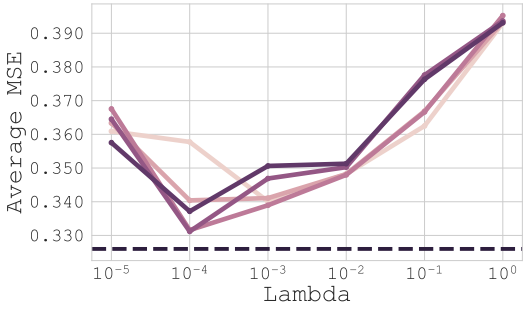
(e) ETTh2 - Horizon 336



(f) Weather - Horizon 336



(g) ETTh2 - Horizon 720



(h) Weather - Horizon 720

Figure 4.5: Results for datasets ETTh2 and Weather on the Transformer baseline, averaged across 3 seeds for each gamma and lambda setting.

strated the existence of an optimal parameter, explicitly computable based on data statistics, which maximizes the benefits of the multi-task framework.

Empirically, our method significantly improved the performance of initially univariate models, such as PatchTST and DLinear, by exploiting the inherent inter-channel dependencies in multivariate data. Remarkably, these models, when regularized with our approach, achieve performance comparable to state-of-the-art multivariate methods such as SAMformer and iTransformer, while maintaining simplicity, interpretability, and computational efficiency.

This work opens promising perspectives for extending our approach to complex nonlinear models, where the final linear structure allows for an efficient application of our theoretical principles. More broadly, this chapter underscores the fundamental importance of considering multivariate forecasting as a multi-task learning problem, encouraging a rethinking of current methods by explicitly integrating inter-channel interactions to achieve better predictive performance.

Chapter 5

On Adapting Foundation Models to Multivariate Time Series Classification

Summary. Foundation models, while highly effective, are often resource-intensive, requiring substantial inference time and memory. This chapter addresses the challenge of making these models more accessible with limited computational resources through meta-channel learning approaches. Our goal is to enable users to run large pre-trained foundation models on standard GPUs without sacrificing performance. We propose a latent space compression strategy that restructures the feature space while preserving essential temporal information. Surprisingly, we show that reducing the latent space to only 2.10% of its original size retains 96.15% of the classification accuracy of the full-sized model. To achieve this, we investigate both classical methods and neural network-based adapters for optimizing multivariate time series representations. Our experiments demonstrate up to a 10× speedup compared to the baseline model without performance degradation, while allowing up to 4.5× more datasets to fit on a single GPU. This enhancement makes foundation models more practical and scalable for real-world applications.

5.1 Introduction

Foundation models have significantly advanced fields such as NLP (Josh Achiam et al., 2023; Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al., 2023b) and computer vision (Dosovitskiy et al., 2021b) by leveraging extensive pre-training on large datasets to create highly adaptable representations. Inspired by these successes, recent research has sought to extend the foundation model paradigm to time series analysis, creating Time Series Foundation Models (TSFMs) (Goswami et al., 2024; Yihang Wang et al., 2024; Garza & Mergenthaler-Canseco, 2023; C. Lin et al., 2024). These models are pre-trained on vast, diverse datasets and then adapted to downstream tasks with minimal additional data, thereby greatly reducing the need for extensive labeled datasets. In contrast, classical approaches for MTS classification, including Dynamic Time Warp-

ing (Salvador & Chan, 2007b), kernel methods, shapelets (Lines, Luke M Davis, et al., 2012), tree-based models (Houtao Deng et al., 2013), and dictionary-based algorithms (J. Lin, Keogh, L. Wei, et al., 2007; J. Lin, Khade, et al., 2012), often struggle with high-dimensional data. Deep learning methods such as ROCKET (Dempster et al., 2020b) and Multi-ROCKET show promise but still fail to address channel interdependencies efficiently.

TSFMs have been proposed for various specialized tasks, including forecasting (Garza & Mergenthaler-Canseco, 2023; Rasul, Arjun Ashok, et al., 2023; Yihang Wang et al., 2024), classification (C. Lin et al., 2024; Feofanov, S. Wen, et al., 2025), and general-purpose modeling (T. Zhou, Peisong Niu, et al., 2023; Goswami et al., 2024). However, a major limitation remains their computational complexity, particularly when applied to multivariate time series containing numerous channels (W. W. Wei, 2018; Bagnall, Dau, et al., 2018b). Current TSFMs typically process channels independently, causing severe memory consumption and excessive runtime, especially under constrained computational resources such as a single standard GPU.

Conversely, we hypothesize that processing multivariate time series channels independently is suboptimal. Some channels may encode redundant information, and it is possible to identify a lower-dimensional set of transformed channels that retain most of the useful signal while reducing computational cost. This reformulation allows us to achieve a significant reduction in inference time and memory footprint, making foundation models more practical for large-scale applications. To test this hypothesis, we introduce dimensionality reduction techniques, enabling TSFMs to operate efficiently without sacrificing predictive performance.

In this chapter, we address these critical computational and scalability issues by introducing dimensionality reduction techniques specifically adapted to TSFMs for multivariate classification tasks. Our objective is twofold: (i) reduce the computational resources required to fine-tune and deploy foundation models, and (ii) preserve classification accuracy through carefully designed adapters that compress the latent feature space effectively. To achieve these goals, we explore various compression methods, including Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Random Projection (Rand Proj), Variance-based Feature Selection (VAR), and neural-network-based linear combiners (lcomb). Unlike traditional dimensionality reduction applied to static data, applying these techniques to temporal data requires careful consideration of temporal dependencies and channel correlations.

We validate our approach extensively on twelve diverse multivariate datasets from the UEA archive (Bagnall, Dau, et al., 2018b), using two prominent TSFMs: MOMENT (Goswami et al., 2024), a large-scale transformer-based model trained with masked reconstruction, and Mantis (Feofanov, S. Wen, et al., 2025), a smaller, contrastively trained vision transformer-based model. Our experiments demonstrate that the dimensionality reduction techniques yield significant computational improvements, including a tenfold inference speedup and the capacity to handle up to 4.5 $\times$ more datasets per GPU, without significant performance loss. Table 5.1 illustrates the severe computational bottlenecks encountered when applying Mantis and MOMENT directly to multivariate datasets without dimensionality reduction,

Table 5.1: Average accuracy over 3 runs under full fine-tuning without an adapter (i.e., using all initial channels).

Model	Duck	Face	Finger	Hand	Heart	Insect	Vowels	Motor	NATOPS	PEMS	Phoneme	SpokeA
Mantis	COM	COM	COM	$0.401_{\pm 0.021}$	COM	COM	$0.981_{\pm 0.005}$	COM	$0.937_{\pm 0.012}$	COM	$0.342_{\pm 0.002}$	$0.987_{\pm 0.001}$
MOMENT	COM	COM	COM	$0.356_{\pm 0.016}$	COM	COM	$0.925_{\pm 0.002}$	COM	TO	COM	TO	TO

as indicated by COM (CUDA Out of Memory error) and TO (Time Out) entries, emphasizing the necessity of our proposed adapters.

5.2 Problem Formulation

Let N denote the number of samples, L the number of time steps, D the number of channels in each multivariate time series, and D' the reduced number of dimensions after applying dimensionality reduction ($D' \leq D$).

Objective. Our goal is to enable efficient multivariate time series classification using pre-trained models while preserving high classification accuracy. We focus on achieving rapid fine-tuning within a 2-hour window on a single GPU without significant performance degradation. To this end, we explore various dimensionality reduction techniques, which preprocess the input data before being processed by foundation models. We then evaluate different fine-tuning strategies to optimize performance under computational constraints.

Challenges. Table 5.1 presents the accuracy results of two TSFMs, Mantis and MOMENT, on a range of multivariate time series datasets under full fine-tuning without the use of any adapter, i.e., without dimensionality reduction. Notably, the results indicate that most of the foundation models encounter severe computational limitations when applied to multivariate data on standard hardware (NVIDIA Tesla V100-32GB GPU), as indicated by COM (CUDA Out of Memory error) and TO (2 hours Time Out) entries. These computational constraints underscore the difficulty of directly applying existing foundation models to multivariate time series with numerous channels, often leading to excessive resource consumption and failures to complete the fine-tuning process. This evidence motivates our exploration of dimensionality reduction techniques, which aim to alleviate these computational bottlenecks and enable foundation models to handle multivariate data more effectively without compromising accuracy.

Problem Definition. Let $\mathbf{X} \in \mathbb{R}^{L \times D}$ denote a multivariate time series with L time steps and D channels, and let $y \in \mathcal{Y} = \{1, \ldots, K\}$ be the corresponding class label for a K-class classification task. We assume that a pre-trained foundation model f encodes each time series channel independently to an embedding vector of size p. Assuming D large, we introduce an adapter that performs latent space compression by mapping the original D channels onto $D' \leq D$ channels to enable efficient processing of high-dimensional data:

$$g : \mathbb{R}^{L \times D} \to \mathbb{R}^{L \times D'}.$$

We consider a set $\mathcal{G}$ of candidate dimensionality reduction techniques (e.g., PCA, trun-

cated SVD, random projection, or neural-network–based linear combiners). The overall classification pipeline is then given by

$$H(\mathbf{X}) = h \circ f \circ g(\mathbf{X}),$$

where $h : \mathbb{R}^{D' \times p} \to \mathcal{Y}$ is a classification head. Our goal is to maximize the classification accuracy under different fine-tuning strategies while respecting a strict resource budget (i.e., fine-tuning must be finished within 2 hours on a single GPU). $\mathbf{X}_i$ denotes the i-th multivariate time series.

Case 1: Head Fine-Tuning

This baseline configuration employs the identity mapping $g_{\text{id}} : \mathbb{R}^{L \times D} \to \mathbb{R}^{L \times D}$, thus passing all D channels directly to the foundation model f. Only the classification head h is fine-tuned, providing a reference scenario without dimension reduction where :

$$H(\mathbf{X}) \;=\; h_\phi \circ f \circ g_{\text{id}}(\mathbf{X}) \;=\; h \circ f(\mathbf{X})$$

Thus, the optimization objective is:

$$\max_{\phi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big(h_\phi \circ f(\mathbf{X}_i) = y_i\Big),$$

and the constraint that this fine-tuning is completed within two hours on a single GPU.

Case 2: Adapter + Head Fine-Tuning

In this setting, the pre-trained foundation model f is kept frozen. The adapter g is parameterized by θ (denoted as g_θ) and the classification head h is parameterized by ϕ. The pipeline is defined as

$$H(\mathbf{X}) = h_\phi \circ f \circ g_\theta(\mathbf{X})$$

The optimization problem is then:

$$\max_{\theta,\,\phi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big(h_\phi \circ f \circ g_\theta(\mathbf{X}_i) = y_i\Big),$$

subject to:

$$g_\theta : \mathbb{R}^{L \times D} \to \mathbb{R}^{L \times D'}, \quad D' \leq D \text{ and } g_\theta \in \mathcal{G}$$

under the same resource constraints.

Case 3: Full Fine-Tuning

In this scenario, the foundation model f is parameterized by ψ and denoted as f_ψ, so that the entire pipeline is fine-tuned. Keeping both the parameterized adapter and head, the pipeline becomes:

$$H(\mathbf{X}) = h_\phi \circ f_\psi \circ g_\theta(\mathbf{X})$$

The corresponding optimization problem is:

$$\max_{\theta,\psi,\phi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big(h_\phi \circ f_\psi \circ g_\theta(\mathbf{X}_i) = y_i\Big),$$

subject to the same mapping constraint:

$$g_\theta : \mathbb{R}^{L\times D} \to \mathbb{R}^{L\times D'}, \quad D' \leq D \text{ and } g_\theta \in \mathcal{G}$$

and the same resource constraint (two hours on a single GPU).

In summary, three distinct approaches are investigated: **(1)** relying on the identity mapping and training only the head , **(2)** freezing f while fine-tuning the adapter and head and **(3)** fully fine-tuning $\{g_\theta, f_\psi, h_\phi\}$. Our primary objective is to reduce channels from D to D' without compromising classification accuracy, while adhering to strict computational limits.

5.3 Proposed Approach

To effectively mitigate computational bottlenecks encountered by foundation models in multivariate time series classification, we propose a latent space compression approach. As shown in Figure 5.1, this approach introduces an adapter function $g_\theta : \mathbb{R}^{L\times D} \to \mathbb{R}^{L\times D'}$ prior to the foundation model f. Our goal is to significantly reduce the dimensionality D of input time series to a smaller latent dimension D', while preserving critical temporal and channel-wise information required for classification. We explore several candidate dimensionality reduction techniques, detailed as follows:

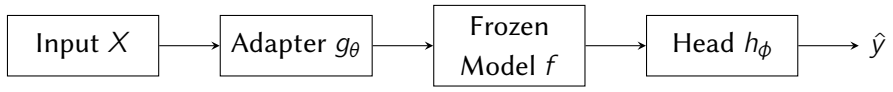
Principal Component Analysis (PCA). seeks to find an orthogonal basis of principal components where a few components capture most of the data's variance. Applying PCA to 3D matrices (N, L, D) poses challenges. A common approach reshapes the data into $(N, T \times D)$ and projects it to $(N, L \times D')$, but this disrupts the temporal structure. Additionally, when $N \ll L \times D$, PCA becomes computationally unstable. To address this, we reshape the data to $(N \times L, D)$, allowing PCA to focus on correlations between channels over all time steps, effectively capturing spatial correlations while preserving temporal information. The learned rotation matrix $\mathbf{W} \in \mathbb{R}^{D'\times D}$ linearly combines the original channels into a lower-dimensional space, applied consistently across all time steps (Pearson, 1901; Jolliffe, 2002).

Truncated Singular Value Decomposition (SVD). Truncated SVD directly decomposes the data without mean-centering, extracting the D' most significant singular components. This approach provides a numerically stable alternative to PCA, effectively capturing the primary structure in the original high-dimensional space (Golub & Van Loan, 2013).

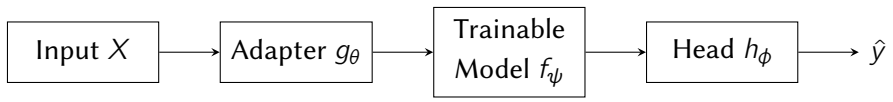
Random Projection (Rand Proj). This approach uses randomly-generated projection matrices to achieve dimensionality reduction efficiently. Unlike PCA or SVD, random projections do not aim to preserve maximum variance but offer rapid computation suitable for large-scale settings (Bingham & Mannila, 2001).

Variance-Based Feature Selection (VAR). VAR retains only the D' channels with the highest variance across samples and time steps, discarding low-variance features assumed less informative for classification (Guyon & Elisseeff, 2003).

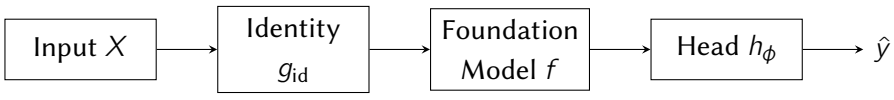
Linear Combiner (lcomb). We propose a learnable neural-network-based adapter, which combines original channels into new meta-channels through a learned weight matrix $\mathbf{W} \in \mathbb{R}^{D \times D'}$. Contrary to classical methods, this approach optimizes channel combinations directly using supervised learning. A top-k variant further sparsifies the matrix $\mathbf{W}$, retaining only the largest k weights per meta-channel for stability and improved generalization.



(a) Frozen f, fine-tune g_θ & h_ϕ.



(b) Fine-tune g_θ, f_ψ & h_ϕ.



(c) No adapter, fine-tune only h_ϕ.

Figure 5.1: Three fine-tuning scenarios in which each adapter g is selected from $\mathcal{G} = \{\text{PCA, Truncated SVD, Rand Proj, VAR, lcomb}\}$.

This comprehensive exploration of adapters aims to identify robust dimensionality reduction methods that significantly lower computational complexity while preserving high accuracy in downstream classification tasks.

Table 5.2: Performance comparison between different adapter configurations for MOMENT and Mantis foundation models with $D' = 5$. The best performance of each adapter+head method is in **bold**; the second best in *italic*. Results for fine-tuning head only given for reference.

Dataset	Model	head	adapter+head					
		no adapter	PCA	SVD	Rand_Proj	VAR	lcomb	lcomb_top_k
DuckDuckGeese	MOMENT	0.460$_{\pm 0.016}$	*0.627*$_{\pm 0.023}$	**0.667**$_{\pm 0.012}$	0.500$_{\pm 0.040}$	0.407$_{\pm 0.012}$	0.427$_{\pm 0.046}$	0.393$_{\pm 0.114}$
	Mantis	0.420$_{\pm 0.020}$	*0.558*$_{\pm 0.023}$	**0.600**$_{\pm 0.032}$	0.487$_{\pm 0.023}$	0.400$_{\pm 0.060}$	0.360$_{\pm 0.020}$	0.393$_{\pm 0.031}$
FaceDetection	MOMENT	0.623$_{\pm 0.006}$	**0.567**$_{\pm 0.002}$	*0.566*$_{\pm 0.001}$	0.552$_{\pm 0.014}$	0.555$_{\pm 0.001}$	TO	TO
	Mantis	0.595$_{\pm 0.004}$	**0.554**$_{\pm 0.001}$	*0.551*$_{\pm 0.007}$	0.533$_{\pm 0.004}$	0.539$_{\pm 0.007}$	0.548$_{\pm 0.008}$	0.550$_{\pm 0.008}$
FingerMovement	MOMENT	0.573$_{\pm 0.012}$	*0.593*$_{\pm 0.032}$	0.573$_{\pm 0.012}$	0.573$_{\pm 0.025}$	**0.613**$_{\pm 0.021}$	0.573$_{\pm 0.032}$	0.540$_{\pm 0.017}$
	Mantis	0.627$_{\pm 0.015}$	**0.593**$_{\pm 0.044}$	0.530$_{\pm 0.030}$	0.570$_{\pm 0.075}$	*0.582*$_{\pm 0.040}$	0.580$_{\pm 0.020}$	0.567$_{\pm 0.046}$
HandMovementDirection	MOMENT	0.401$_{\pm 0.008}$	*0.410*$_{\pm 0.043}$	0.365$_{\pm 0.036}$	0.405$_{\pm 0.041}$	0.369$_{\pm 0.039}$	0.378$_{\pm 0.047}$	**0.414**$_{\pm 0.008}$
	Mantis	0.342$_{\pm 0.021}$	**0.396**$_{\pm 0.021}$	*0.351*$_{\pm 0.089}$	0.329$_{\pm 0.083}$	0.329$_{\pm 0.031}$	0.320$_{\pm 0.034}$	0.320$_{\pm 0.028}$
Heartbeat	MOMENT	0.740$_{\pm 0.003}$	0.732$_{\pm 0.000}$	0.732$_{\pm 0.005}$	**0.756**$_{\pm 0.005}$	0.725$_{\pm 0.006}$	*0.737*$_{\pm 0.005}$	*0.737*$_{\pm 0.013}$
	Mantis	0.811$_{\pm 0.010}$	0.766$_{\pm 0.005}$	0.737$_{\pm 0.012}$	0.776$_{\pm 0.013}$	**0.780**$_{\pm 0.010}$	0.748$_{\pm 0.006}$	*0.779*$_{\pm 0.014}$
InsectWingbeat	MOMENT	0.284$_{\pm 0.003}$	**0.239**$_{\pm 0.003}$	*0.224*$_{\pm 0.003}$	0.193$_{\pm 0.027}$	0.195$_{\pm 0.004}$	0.167$_{\pm 0.014}$	0.213$_{\pm 0.010}$
	Mantis	0.614$_{\pm 0.005}$	0.344$_{\pm 0.013}$	*0.352*$_{\pm 0.010}$	0.333$_{\pm 0.035}$	0.238$_{\pm 0.012}$	0.171$_{\pm 0.013}$	**0.354**$_{\pm 0.041}$
JapaneseVowels	MOMENT	0.885$_{\pm 0.002}$	0.801$_{\pm 0.009}$	*0.803*$_{\pm 0.003}$	0.796$_{\pm 0.011}$	0.734$_{\pm 0.008}$	0.797$_{\pm 0.035}$	**0.819**$_{\pm 0.027}$
	Mantis	0.979$_{\pm 0.006}$	**0.922**$_{\pm 0.009}$	0.897$_{\pm 0.012}$	*0.902*$_{\pm 0.008}$	0.885$_{\pm 0.010}$	0.798$_{\pm 0.070}$	0.816$_{\pm 0.027}$
MotorImagery	MOMENT	0.643$_{\pm 0.015}$	0.590$_{\pm 0.010}$	**0.607**$_{\pm 0.012}$	0.567$_{\pm 0.032}$	0.550$_{\pm 0.010}$	0.583$_{\pm 0.015}$	*0.593*$_{\pm 0.025}$
	Mantis	0.600$_{\pm 0.036}$	*0.593*$_{\pm 0.025}$	0.590$_{\pm 0.017}$	0.577$_{\pm 0.029}$	**0.607**$_{\pm 0.025}$	0.557$_{\pm 0.045}$	**0.607**$_{\pm 0.055}$
NATOPS	MOMENT	0.872$_{\pm 0.011}$	*0.776*$_{\pm 0.008}$	0.739$_{\pm 0.017}$	0.774$_{\pm 0.032}$	**0.813**$_{\pm 0.020}$	0.596$_{\pm 0.017}$	0.769$_{\pm 0.031}$
	Mantis	0.944$_{\pm 0.011}$	**0.874**$_{\pm 0.014}$	0.820$_{\pm 0.012}$	*0.852*$_{\pm 0.038}$	0.850$_{\pm 0.035}$	0.787$_{\pm 0.003}$	0.826$_{\pm 0.036}$
PEMS-SF	MOMENT	0.834$_{\pm 0.026}$	0.678$_{\pm 0.007}$	0.511$_{\pm 0.022}$	0.644$_{\pm 0.027}$	0.611$_{\pm 0.015}$	**0.740**$_{\pm 0.010}$	*0.697*$_{\pm 0.013}$
	Mantis	0.923$_{\pm 0.023}$	**0.674**$_{\pm 0.032}$	*0.640*$_{\pm 0.045}$	0.615$_{\pm 0.023}$	0.615$_{\pm 0.055}$	0.584$_{\pm 0.025}$	0.594$_{\pm 0.065}$
PhonemeSpectra	MOMENT	0.234$_{\pm 0.001}$	*0.234*$_{\pm 0.002}$	0.212$_{\pm 0.002}$	**0.245**$_{\pm 0.003}$	0.228$_{\pm 0.004}$	TO	TO
	Mantis	0.296$_{\pm 0.003}$	0.270$_{\pm 0.003}$	0.259$_{\pm 0.001}$	*0.293*$_{\pm 0.002}$	**0.294**$_{\pm 0.004}$	0.279$_{\pm 0.002}$	0.286$_{\pm 0.001}$
SpokenArabicDigits	MOMENT	0.977$_{\pm 0.001}$	*0.972*$_{\pm 0.000}$	**0.978**$_{\pm 0.000}$	0.961$_{\pm 0.008}$	0.935$_{\pm 0.002}$	TO	TO
	Mantis	0.940$_{\pm 0.003}$	**0.962**$_{\pm 0.003}$	0.933$_{\pm 0.001}$	0.879$_{\pm 0.004}$	*0.946*$_{\pm 0.003}$	0.834$_{\pm 0.019}$	0.873$_{\pm 0.019}$
Avg Ratio to head only	MOMENT	1.000	0.973	0.939	0.930	0.893	0.870	0.904
	Mantis	1.000	0.950	0.920	0.900	0.882	0.823	0.875

5.4 Experimental Evaluation

Experimental Setup. All experiments were conducted using an NVIDIA Tesla V100 GPU (32GB) with a strict runtime constraint of two hours per fine-tuning task. Models exceeding these limits are reported as either COM (CUDA Out-of-Memory) or TO (Time-Out).

Foundation Models. We evaluate two representative TSFMs:

- MOMENT (Goswami et al., 2024): A large-scale transformer-based model pre-trained via masked reconstruction (341M parameters).
- Mantis (Feofanov, S. Wen, et al., 2025): A smaller Vision Transformer (ViT)-based

model pre-trained via contrastive learning (8M parameters).

Datasets. This study draws on 12 UEA datasets (Bagnall, Dau, et al., 2018b), each containing at least 10 channels, to ensure that dimensionality reduction (from D to D') confers a tangible advantage. The UEA archive comprises 30 multivariate datasets, but those with fewer than 10 channels generally derive limited benefit from such reduction. While our method is applicable to any D, it provides the greatest impact when D is sufficiently large. The experimental results presented in this work are based on a diverse set of datasets, whose main characteristics are summarized in Table 5.3. These datasets span a variety of domains and tasks, offering a comprehensive evaluation of the fine-tuning methods under consideration. For instance, the datasets include time-series data from physiological measurements (e.g., *Heartbeat*, *MotorImagery*), sensor readings (e.g., *PEMS-SF*), and acoustic signals (e.g., *PhonemeSpectra*, *SpokenArabicDigits*). The number of channels, sequence lengths, and class distributions vary significantly across datasets, ensuring that the results generalize across different data modalities and problem settings. In the case of the *InsectWingbeat* dataset, we specifically subsampled 1000 examples from the original training set (which contains 30,000 examples) and 1000 from the original test set (of 20,000 examples) to reduce computational overhead while maintaining sufficient variety in the data for robust model evaluation. Each dataset was carefully chosen to challenge the models across different feature spaces, class imbalances, and temporal dependencies. For example, the *JapaneseVowels* dataset focuses on speaker classification based on vowel sounds, while the *DuckDuckGeese* dataset involves distinguishing animal sounds with varying levels of complexity in terms of sequence length and channel dimensionality. By including these datasets, we ensure that the evaluation framework captures the performance of fine-tuning methods across a wide spectrum of classification tasks.

Table 5.3: Main characteristics of the considered datasets.

Dataset	Train Size	Test Size	# of channels	Sequence Len	# of classes
DuckDuckGeese (Duck)	60	40	1345	270	5
FaceDetection (Face)	5890	3524	144	62	2
FingerMovements (Finger)	316	100	28	50	2
HandMovementDirection (Hand)	320	147	10	400	4
Heartbeat (Heart)	204	205	61	405	2
InsectWingbeat (Insect)	1000	1000	200	78	10
JapaneseVowels (Vowels)	270	370	12	29	9
MotorImagery (Motor)	278	100	64	3000	2
NATOPS	180	180	24	51	6
PEMS-SF (PEMS)	267	173	963	144	7
PhonemeSpectra (Phoneme)	3315	3353	11	217	39
SpokenArabicDigits (SpokeA)	6599	2199	13	93	10

Definitions. We now define the terms "head" and "adapter," which are subsequently used depending on the fine-tuning scenario. The head is a linear classification layer added to the output of the foundation model, while the adapter is inserted upstream of the foundation model.

Full Fine-Tuning Regime. We first show that full fine-tuning without adapters leads to errors such as Time-Out (TO) or Cuda Out of Memory (COM), given our computational and time constraints. On the few datasets that meet our requirements, we compared the results of full fine-tuning without adapters with those of head-only fine-tuning with an adapter to assess whether fine-tuning the entire foundation model is worthwhile. The results indicate that it is not. For the *Hand* dataset, the average performance is 0.401 for Mantis and 0.356 for MOMENT with full fine-tuning without adapters, whereas head-only fine-tuning yields 0.401 and 0.342, respectively, with a high variance of 0.02 for MOMENT across different seeds. This clearly demonstrates that full fine-tuning is unnecessary for this dataset. Finally, on the *Vowels* dataset—the second and final dataset for which both models meet our requirements—Mantis and MOMENT achieve performances of 0.981 and 0.925, respectively, with full fine-tuning without adapters, and 0.979 and 0.885 when fine-tuning only the head. While Mantis shows similar performance under both regimes, full fine-tuning slightly outperforms head-only fine-tuning for MOMENT. However, given that MOMENT consists of 341M parameters, full fine-tuning incurs a significant computational cost, which is reflected in the fact that out of the 12 datasets considered, only two meet our requirements for MOMENT without errors as shown in Table 5.1.

Head Only vs Adapter+Head. We subsequently focus on comparing head-only fine-tuning with adapter+head fine-tuning. While full fine-tuning is prohibitively expensive, and the trade-off between computational resources and results appears to favor head-only fine-tuning, why then consider adapter+head fine-tuning? The answer is twofold: not only does adapter+head fine-tuning preserve the baseline performance of head-only fine-tuning at an average of 97.15% across both models, but it also reduces the fine-tuning time by 10× for MOMENT and approximately 2× for Mantis. In summary, we first demonstrate that full fine-tuning without adapters is too costly, making head-only fine-tuning the preferable option when comparing these two approaches. We then show that there is no statistically significant difference in performance between head-only fine-tuning and adapter+head fine-tuning, as evidenced by the statistical tests in Figure 5.2, which reveal a single cluster encompassing all adapter methods as well as the "No Adapter" configuration. Meanwhile, the running time is drastically reduced, as highlighted in Figure 5.3. Detailed quantitative results are presented in the following paragraph.

Results. We present an experimental comparison of multiple adapters when fine-tuning both the adapter and the head of a foundation model. We evaluate MOMENT and Mantis on twelve multivariate time series datasets from the UEA archive with more than ten features, reducing the data from an average of 240 channels to 5. Rather than discarding

channels, these adapters construct new *metachannels* through linear or nonlinear transformations, thereby retaining important information in a significantly lower-dimensional space ($\frac{5}{240} \approx 2.08\%$ of the original dimension). Remarkably, the PCA-based adapter preserves on average 97.30% of the accuracy of the no-adapter configuration for MOMENT and 95.00% for Mantis, despite this drastic dimensionality reduction (see Table 5.2).

We also report results for the baseline scenario in which only the classification head is fine-tuned (i.e., without any adapter). As shown in Table 5.2, accompanied by statistical tests in Figure 5.2, there is no statistically significant difference among the methods on average over all datasets, including the head-only baseline. Nevertheless, Figure 5.3 demonstrates that using adapters drastically reduces computation time: for MOMENT, they are on average over ten times faster than the no-adapter setting, and for Mantis, they yield a two-fold speedup. An exception is the Linear Combiner (lcomb) adapter, a deep learning-based model that requires invoking the foundation model at each fine-tuning step. In contrast, other (non-deep) methods only transform the data *once* into embeddings, then train the classification head without repeatedly running the foundation model. This significantly reduces runtime compared to approaches such as lcomb.

Notably, Table 5.2 indicates that the no-adapter strategy outperforms on certain datasets, suggesting that the best dimensionality may vary per dataset. Consequently, more sophisticated adapters may be needed for robust dimension reduction in challenging cases.

Finally, by comparing results in full fine-tuning results with Table 5.1, we now observe that lcomb now enables fine-tuning on 12/12 datasets for Mantis and 9/12 for MOMENT on a single GPU, whereas full fine-tuning only accommodated 5 and 2 datasets, respectively. This corresponds to a 2.4× increase for Mantis and a 4.5× increase for MOMENT in terms of the number of datasets that fit in a single GPU, within two hours.

Table 5.4: Performance comparison between fine-tuning methods with different adapter configurations for the MOMENT foundation model

Dataset	adapter+head			
	PCA	Scaled PCA	Patch_8	Patch_16
DuckDuckGeese	$0.667_{\pm 0.012}$	$0.533_{\pm 0.031}$	$0.567_{\pm 0.031}$	$0.573_{\pm 0.031}$
FaceDetection	$0.566_{\pm 0.001}$	COM	$0.582_{\pm 0.003}$	$0.558_{\pm 0.004}$
FingerMovement	$0.573_{\pm 0.012}$	$0.563_{\pm 0.032}$	$0.633_{\pm 0.012}$	$0.563_{\pm 0.015}$
HandMovementDirection	$0.365_{\pm 0.036}$	$0.356_{\pm 0.043}$	$0.464_{\pm 0.021}$	$0.383_{\pm 0.021}$
Heartbeat	$0.732_{\pm 0.005}$	$0.728_{\pm 0.003}$	$0.738_{\pm 0.007}$	$0.741_{\pm 0.013}$
InsectWingbeat	$0.224_{\pm 0.003}$	$0.239_{\pm 0.003}$	$0.458_{\pm 0.002}$	$0.459_{\pm 0.004}$
JapaneseVowels	$0.803_{\pm 0.003}$	$0.723_{\pm 0.020}$	$0.967_{\pm 0.002}$	$0.963_{\pm 0.002}$
MotorImagery	$0.607_{\pm 0.012}$	$0.590_{\pm 0.020}$	$0.577_{\pm 0.006}$	$0.597_{\pm 0.015}$
NATOPS	$0.739_{\pm 0.017}$	$0.731_{\pm 0.012}$	$0.857_{\pm 0.003}$	$0.915_{\pm 0.003}$
PEMS-SF	$0.511_{\pm 0.022}$	$0.678_{\pm 0.007}$	$0.719_{\pm 0.012}$	$0.696_{\pm 0.018}$
PhonemeSpectra	$0.212_{\pm 0.002}$	$0.227_{\pm 0.008}$	$0.224_{\pm 0.001}$	$0.186_{\pm 0.001}$
SpokenArabicDigits	$0.978_{\pm 0.000}$	$0.963_{\pm 0.001}$	$0.967_{\pm 0.001}$	$0.956_{\pm 0.001}$

Table 5.5: Performance comparison between fine tuning methods with different adapter configurations for Mantis foundation model

Dataset	adapter+head			
	PCA	Scaled PCA	Patch_8	Patch_16
DuckDuckGeese	$0.558_{\pm 0.023}$	$0.522_{\pm 0.023}$	$0.467_{\pm 0.031}$	$0.440_{\pm 0.035}$
FaceDetection	$0.554_{\pm 0.001}$	$0.550_{\pm 0.010}$	$0.551_{\pm 0.003}$	$0.547_{\pm 0.007}$
FingerMovement	$0.593_{\pm 0.044}$	$0.583_{\pm 0.023}$	$0.530_{\pm 0.036}$	$0.570_{\pm 0.053}$
HandMovementDirection	$0.367_{\pm 0.042}$	$0.327_{\pm 0.056}$	$0.396_{\pm 0.021}$	$0.369_{\pm 0.021}$
Heartbeat	$0.736_{\pm 0.010}$	$0.734_{\pm 0.014}$	$0.766_{\pm 0.005}$	$0.763_{\pm 0.018}$
InsectWingbeat	$0.344_{\pm 0.013}$	$0.268_{\pm 0.005}$	$0.287_{\pm 0.011}$	$0.266_{\pm 0.006}$
JapaneseVowels	$0.890_{\pm 0.008}$	$0.865_{\pm 0.016}$	$0.922_{\pm 0.009}$	$0.921_{\pm 0.011}$
MotorImagery	$0.567_{\pm 0.006}$	$0.552_{\pm 0.045}$	$0.593_{\pm 0.025}$	$0.573_{\pm 0.065}$
NATOPS	$0.837_{\pm 0.012}$	$0.840_{\pm 0.017}$	$0.874_{\pm 0.014}$	$0.870_{\pm 0.008}$
PEMS-SF	$0.584_{\pm 0.010}$	$0.613_{\pm 0.025}$	$0.634_{\pm 0.013}$	$0.674_{\pm 0.032}$
PhonemeSpectra	$0.270_{\pm 0.003}$	$0.262_{\pm 0.008}$	$0.234_{\pm 0.002}$	$0.205_{\pm 0.006}$
SpokenArabicDigits	$0.962_{\pm 0.003}$	$0.952_{\pm 0.003}$	$0.921_{\pm 0.006}$	$0.899_{\pm 0.002}$

5.5 Qualitative Study

Hyperparameter Sensitivity of PCA. In this experiment, we implemented a variant of PCA called Patch PCA. Unlike the traditional approach where the input time series of shape (N, L, D) is reshaped into $(N \times L, D)$ before applying PCA, our method reshapes the input into $(N \times n_p, pws \times D)$, where n_p represents the number of patches in the sequence and pws refers to the patch window size. The case where $pws = 1$ corresponds to the standard PCA approach. We compare the results across different patch window sizes ($pws = 1, 8, 16$). These experiments show no clear pattern in performance across the different patch sizes, suggesting that the patch window size can be treated as a hyperparameter to be tuned based on the specific dataset. Furthermore, we introduced two key hyperparameters for our PCA implementation: the patch window size (pws) and the option to scale the data before performing PCA. The results of PCA presented in Tables 5.4 and 5.5 reflect the accuracy obtained for each configuration of these two hyperparameters, allowing us to explore the impact of different settings on performance and to choose the best hyperparameters to present the results in Table 5.2. This flexibility in the PCA configuration allows us to adapt the method to a wide range of tasks, optimizing both performance and computational efficiency.

Hyperparameter Sensitivity of lcomb. In addition to the standard *lcomb* configuration, we evaluated a variant called *lcomb_top_k*, which introduces a form of regularization to make the attention mechanism more stable. In *lcomb_top_k*, only the top k largest attention weights are selected, and each row of the attention matrix is rescaled by dividing by the sum of these k weights. For our experiments, we set $k = 7$. This mechanism is designed to reduce noise in the attention distribution, focusing the model on the most im-

portant relationships between elements in the input. The results shown in Figure 5.4 show the performance comparison between *lcomb* and *lcomb_top_k* across several datasets for both MOMENT and Mantis foundation models.

5.6 Tests and Comparisons

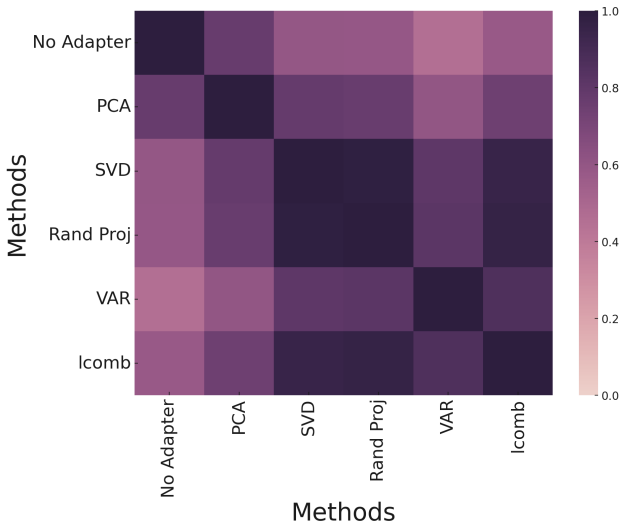
Statistical Tests. The heatmap shown in Figure 5.2 presents the pairwise p-values between different fine-tuning methods applied to the MOMENT and Mantis foundation models across several datasets. The methods compared include *No Adapter*, *PCA*, *SVD*, *Rand Proj*, *VAR*, and *lcomb*. The p-values were calculated using a two-sample Student's t-test with unequal variances, based on accuracy results obtained from three different seeds for each method. The null hypothesis for each comparison states that there is no significant difference in the mean performance, in terms of accuracy, between the two methods being compared. A p-value close to 1 supports this hypothesis, indicating that the two methods yield statistically similar performance. In contrast, a p-value close to 0 suggests a significant difference. In the MOMENT heatmap, the lowest p-value observed is 0.46, while for Mantis, the minimum p-value is 0.25. These visualizations indicate that there is no statistically significant difference between fine-tuning using adapter + head with different adapters, and similarly, no difference is observed between adapter + head and head-only fine-tuning, regardless of the adapter used.

Rank comparisons. Figure 5.5 shows a comparison of the average rank for different adapter methods used in the MOMENT and Mantis foundation models. The average ranks were computed across all datasets and averaged over three seeds. The comparison gives insight into the relative performance of each adapter method when applied to these two models. For the MOMENT foundation model, as depicted in Figure 5.5a, the *PCA* adapter ranks the lowest, indicating the best performance, while the *lcomb* adapter ranks the highest, showing relatively lower performance. The remaining adapters—*SVD*, *Rand_Proj*, and *VAR*—lie in between, with *Rand_Proj* and *SVD* showing close performance. Similarly, in the case of the Mantis foundation model (Figure 5.5b), *PCA* exhibits the lowest average rank, implying superior performance. *Rand_Proj* also performs relatively worse in this case. The consistency of PCA's superior performance across both models highlights its effectiveness.

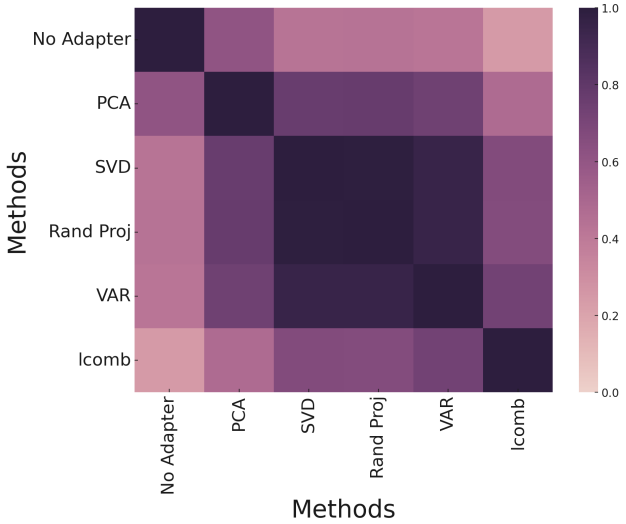
5.7 Conclusions

We presented a latent space compression framework that preserves 96.15% of baseline accuracy while retaining $\sim$ 2% of the original embedding dimensions, yielding up to a 10$\times$ speedup and enabling 4.5$\times$ more datasets per GPU. These gains demonstrate the effectiveness of adapters in scaling foundation models under limited resources. Future

directions include refining compression techniques and extending the approach to more diverse time series domains.

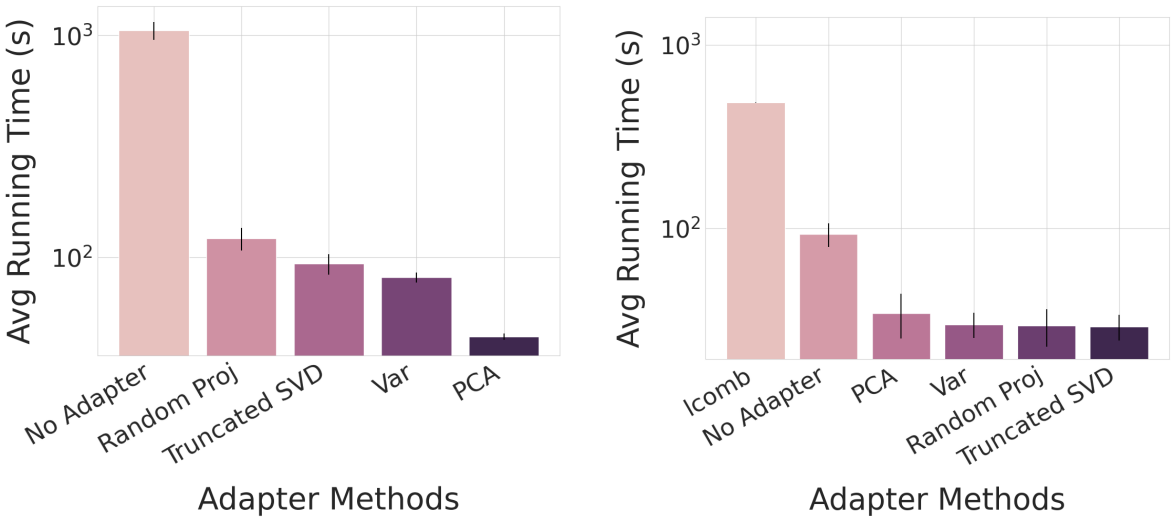


(a) Heatmap of Pairwise p-values for Adapter Methods for MOMENT Foundation Model



(b) Heatmap of Pairwise p-values for Adapter Methods for Mantis Foundation Model

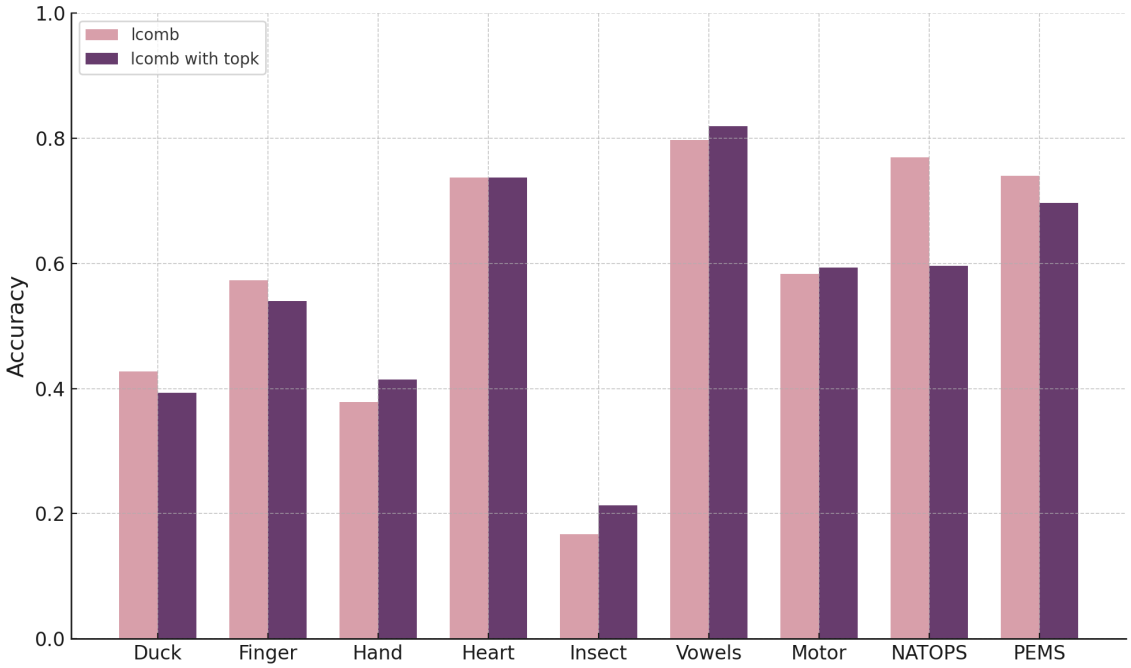
Figure 5.2: Heatmap of pairwise p-values for adapter methods applied to MOMENT and Mantis foundation models, averaged across all datasets and three different seeds. "No adapter" refers to fine-tuning the head only, while applying a dimensionality reduction technique corresponds to fine-tuning both the adapter and the head. The results indicate no statistically significant difference in performance between the no-adapter setting (i.e., using all D channels for head fine-tuning) and the adapter-based approach (i.e., reducing to D' channels before fine-tuning). All performance results are detailed in Table 5.2. However, while performance remains unchanged, adapter-based methods significantly reduce runtime, as shown in Figure 5.3.



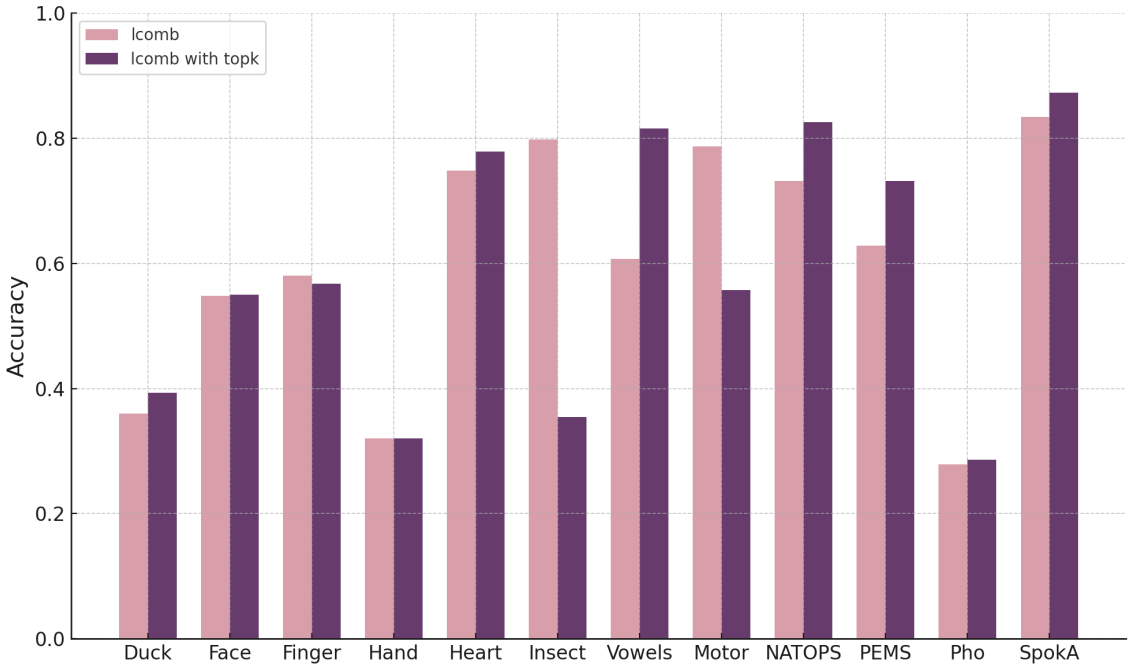
(a) MOMENT

(b) Mantis

Figure 5.3: Comparison of running times for MOMENT and Mantis models, averaged across all datasets and three different seeds. For MOMENT, which has 341M parameters, using an adapter reduces the running time by approximately 10× compared to the version without an adapter, while retaining 97.30% of the original performance (see Table 5.2). For Mantis, a significantly smaller model with around 8M parameters, the running time is also reduced by a factor of 2 when using a PCA-based adapter, while maintaining 95% of the original performance. "No Adapter" means fine-tuning the head only.

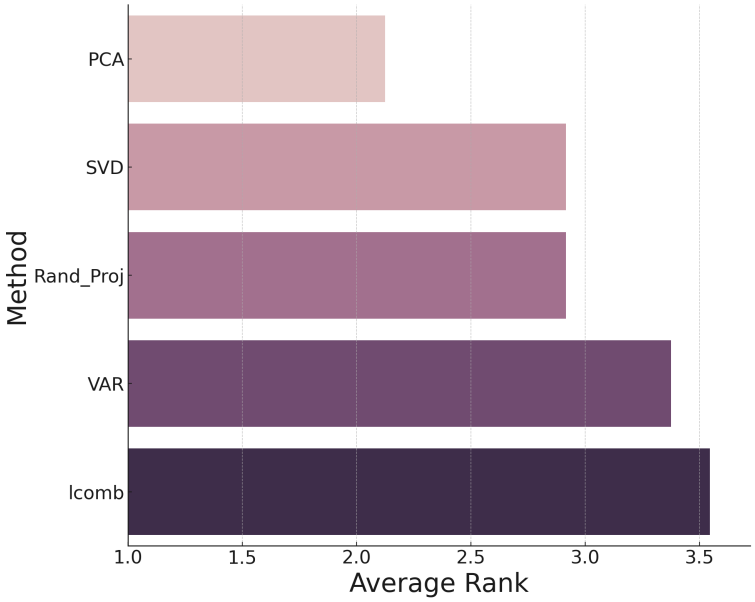


(a) MOMENT

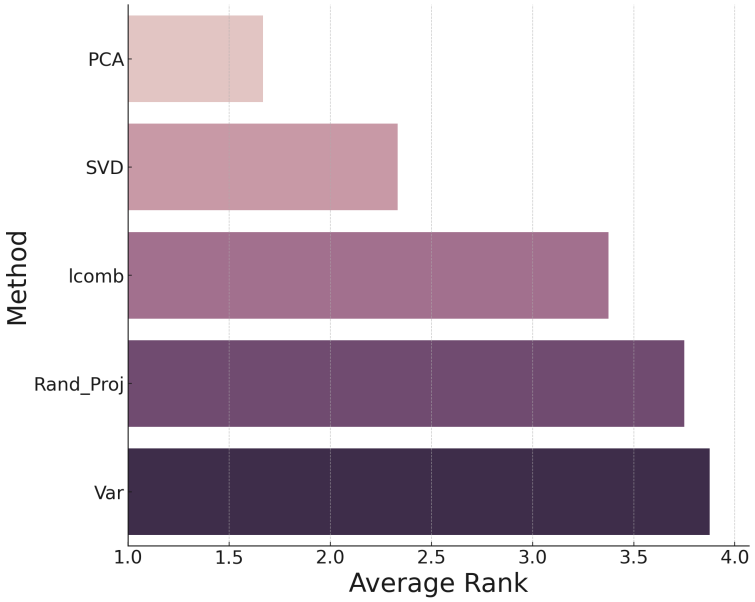


(b) Mantis

Figure 5.4: Performance Comparison between *lcomb* and *lcomb_top_k* configurations for both MOMENT and Mantis models.



(a) Adapter's Average Rank for MOMENT Foundation Model



(b) Adapter's Average Rank for Mantis Foundation Model

Figure 5.5: Comparison of Adapter's Average Rank for MOMENT and Mantis Foundation Models averaged across all datasets and three different seeds

Chapter 6

Conclusions and Future Work

6.1 Improvement of multivariate time series representations

This thesis set out to develop efficient, interpretable, and robust representations for multivariate time series by embracing the inherent interdependencies among channels rather than treating each independently. Our approach was initiated with the creation of SAMformer, a shallow transformer architecture tailored for multivariate data. SAMformer distinguishes itself by employing a channel-wise attention mechanism, which reduces the number of parameters compared to timestamp-level attention and consequently mitigates overfitting. This design not only enables the extraction of clearer, block-structured attention matrices that reveal correlations among features, but it also provides significant interpretability advantages. For instance, in datasets like Weather, where physical quantities such as pressure, humidity, and temperature exhibit stable temporal correlations due to their underlying physical laws, the channel attention mechanism proved exceptionally effective. In non-physical domains such as inter-city traffic data, this method maintained its robustness and efficacy.

In addition, we integrated sharpness-aware minimization (SAM) into the training process to steer the optimization away from suboptimal local minima. This enhanced the stability and generalization of our model, ensuring that the learned representations were both robust and capable of capturing nuanced temporal dynamics. The simplicity of our SAMformer, combined with the careful application of SAM, demonstrated that even shallow architectures can address the limitations of standard transformers, which often suffer from excessive regularization and an overly large number of parameters.

Beyond this contribution, we also sought to understand the behavior of forecasting networks from a theoretical perspective. In this part of the thesis, we examined the theoretical foundations of multivariate time series forecasting through the lens of random matrix theory and the concentration of measure phenomenon. We focused on the linear

projection layer that typically follows the nonlinear feature extractor in deep forecasting architectures. By assuming Lipschitz continuity of the nonlinear mapping, we showed that the resulting representation does not drastically distort the input distribution, allowing for a tractable theoretical analysis.

Building on this framework, we derived an analytical understanding of the learning dynamics of linear predictors applied to deep features. Our analysis provides precise conditions under which these predictors generalize well, and explains the impact of architectural and statistical properties on learning curves. Notably, the theory accounts for the observed empirical performance of deep forecasting models, whose behavior aligns closely with the theoretical predictions.

This theoretical contribution bridges the gap between complex nonlinear models and simpler linear approximations, offering both interpretability and analytical tractability. It also suggests promising avenues for principled model design and regularization strategies in high-dimensional time series forecasting, grounded in rigorous mathematical theory.

Furthermore, we explored the adaptation of foundation models through dimensionality reduction techniques. By reducing the latent space from a high-dimensional space D to a more manageable D', our approach ensured that each meta-channel incorporated information from multiple original channels. Even when applying straightforward methods such as Principal Component Analysis (PCA), we achieved impressive latent space compression and maintained high forecasting accuracy, scalability, and practicality. This demonstrates that simple yet effective dimensionality reduction techniques can serve as a foundation for more advanced models in the future, potentially leading to the development of large-scale foundation models for time series analysis.

In essence, this thesis shows that robust and interpretable multivariate time series representations can be achieved through the thoughtful integration of simple model architectures, effective optimization techniques, and rigorous theoretical analysis. By unifying these elements, we have established a framework that addresses the challenges of overfitting and high dimensionality and enhances the interpretability and efficiency of forecasting models. We hope that the insights and methods presented herein will lay the groundwork for future research, inspiring the development of even more sophisticated yet computationally tractable models that fully exploit the richness of multivariate time series data.

6.2 Open Problems

This section presents a comprehensive overview of potential directions, organized as a single coherent discussion to highlight their interconnections and how they might collectively inform the next generation of approaches in this field.

6.2.1 Hybrid Channel-Temporal Attention

A first line of investigation concerns the contexts in which *channel attention* is most effective and when it might be beneficial to blend channel-level attention with *temporal attention*. We have seen that channel attention excels in scenarios where the correlations between different channels are either stable or relatively slow-varying over time, as is often the case with physical quantities (e.g., temperature, humidity, and pressure in meteorological data). However, in real-world applications, especially those involving social or economic data, correlations may evolve rapidly, rendering static channel correlations insufficient. Future work could thus explore dynamic or adaptive channel attention mechanisms that track changes in inter-channel dependencies over time, potentially in tandem with conventional temporal attention. By designing a hybrid attention scheme, one might capture not only the slow changes in physical relationships but also abrupt shifts in more volatile contexts (such as financial indicators or complex traffic patterns).

6.2.2 Sharpness-Aware Minimization: Data and Model Scaling

A second topic of considerable interest lies in investigating how the performance gains offered by *SAM* relate to both the *quantity* and the *quality* of the available data. The empirical results obtained so far suggest that SAM is particularly valuable in lower-data regimes, where it helps the model find flatter minima and avoid overfitting. Yet, one might ask whether these advantages hold when large-scale datasets become available, or when data noise and inconsistencies are relatively high. It is plausible that the benefits of SAM might diminish if the model already has ample data to learn stable representations. A systematic evaluation of SAM's efficacy across a spectrum of data sizes, from extremely sparse to very large datasets, could yield actionable insights on when, and how, SAM should be employed. Moreover, it may be instructive to assess whether *deep* networks (e.g., deeper transformer stacks or other architectures) continue to profit from SAM, or if the magnitude of gains diminishes as model capacity grows and data becomes more abundant.

6.2.3 Anisotropic Sharpness-Aware Minimization

Another promising extension is to *develop an improved version of SAM* that is sensitive to anisotropic directions of sharpness in the parameter space. Presently, SAM applies isotropic perturbations by seeking a worst-case loss within a uniform ball around the parameters, which can be suboptimal if only a few directions in the parameter space exhibit high curvature. In other words, SAM in its current form addresses sharpness in a first-order manner, focusing on the largest eigenvalue of the Hessian but not explicitly accounting for other eigenvalues that might also be substantial. A more advanced, anisotropic approach could weigh perturbations by the eigenvalues of the Hessian, effectively shaping the perturbation region as an ellipse rather than a sphere. This would allow the optimization to focus more directly on the directions that truly matter for sharpness, potentially improv-

ing training stability and convergence. However, implementing this approach in practice requires careful consideration of computational overhead, as approximating or computing the Hessian in high-dimensional models can be extremely resource-intensive. Still, even partial or low-rank approximations to the Hessian might confer substantial benefits, pointing to an exciting area of algorithmic research.

6.2.4 Scaling SAMformer as a Foundation Model

Beyond improvements in optimization strategies, it would be worthwhile to consider how SAMformer itself could be *scaled up or adapted* to serve as a foundation model. Currently, SAMformer stands out for its shallow depth and channel-focused attention, which make it computationally efficient and interpretable. Future research might explore ways of extending it into deeper or broader architectures that remain tractable for large-scale training. For instance, one could imagine a multi-stage transformer that progressively refines channel interactions, or a variant that includes a learned embedding layer capable of handling heterogeneous data types (e.g., categorical channels alongside continuous signals). Additionally, as foundation models are increasingly employed for tasks beyond forecasting—such as classification, anomaly detection, and imputation—there is ample opportunity to generalize SAMformer to these domains. By carefully incorporating channel-level representations, one might preserve the interpretability and robustness that shallow transformers have demonstrated, even when scaling to much larger parameter counts or more diverse datasets.

6.2.5 Theoretical Insights into Rank and Entropy Collapse

From a *theoretical perspective*, there remain several open questions that could be fruitfully investigated. One line of inquiry involves the interplay between *rank collapse* and *entropy collapse* in attention matrices. Our experiments and theoretical arguments have suggested that rank collapse—where the attention matrix degenerates toward rank one—can severely degrade performance in multivariate time series tasks, more so than the entropy collapse that is often cited in natural language processing and computer vision. It would be valuable to compare these phenomena across different domains systematically, thereby determining whether certain tasks or data modalities are more susceptible to rank collapse than others. Further theoretical developments could also clarify whether techniques like SAM implicitly act as a form of nuclear norm maximization (i.e., encouraging higher-rank solutions in the attention matrix). Such an understanding would be beneficial in bridging the gap between time series, NLP, and computer vision, potentially revealing common underlying principles.

6.2.6 Extending Multi-Task Regularization Theory to Nonlinear Models

A promising theoretical extension involves generalizing our *multi-task regularization* framework and the *concentrated random vector assumption* from linear settings to nonlinear neural networks. Previously, we derived closed-form expressions for optimal hyperparameters in linear models and validated their accuracy empirically. Interestingly, preliminary experiments on nonlinear architectures, such as shallow transformers, exhibited learning curves closely matching linear predictions, suggesting that the underlying theoretical principles could extend beyond linearity.

Formalizing this observation might leverage the *concentration hypothesis*, which posits that due to nonlinearities and Lipschitz continuity, neural networks typically do not drastically alter the distribution of their inputs. Specifically, if intermediate layers exhibit sufficient contraction properties, the network's output distribution remains close to that of the input. In forecasting models, the final linear projection layer would then naturally lend itself to analysis using our existing linear theoretical framework. By validating these assumptions, we could extend rigorous hyperparameter selection and establish robust error bounds for a broad class of deep forecasting architectures.

6.2.7 Dimension Reduction and Latent Space Representation

Dimension reduction and foundation models offer yet another domain ripe for future exploration. Our initial experiments showed that even basic techniques like PCA can compress the latent space of multivariate time series without substantial loss in accuracy. Nevertheless, more sophisticated methods—such as manifold learning or autoencoder-based dimensionality reduction—could be investigated to capture nonlinear structures in the data. Beyond the compression ratio itself, interpretability stands as a key consideration: how can we combine channels in a way that remains transparent to domain experts? For instance, if two channels contain redundant information, should they be merged early in the architecture, or is it preferable to let the model learn such correlations automatically? Studying the effect of channel redundancy on both model performance and training stability could yield important guidelines for the design of future foundation models. Additionally, practical issues arise from overlapping subsequences during training: large overlaps can lead to repetitive data samples that may skew the optimization process. Investigating how to optimally segment or sample multivariate time series might further refine the performance and computational efficiency of these methods.

6.2.8 Computational Challenges of 4D Attention

In a more general sense, *4D attention mechanisms*—where each time step of a given channel attends to every time step of every other channel—represent an intriguing but potentially

computationally explosive direction. Such a fully-coupled scheme would likely demand advanced techniques such as flash attention, low-rank factorization, or additional dimension reduction methods to remain tractable. Balancing expressiveness, interpretability, and efficiency in this four-dimensional space is a non-trivial challenge, but it holds promise for capturing intricate interdependencies between channels and time steps that simpler architectures might miss.

6.2.9 Multivariate Time Series Forecasting with Multi-Scale Spatio-Temporal Disentanglement

A promising strategy for handling multivariate time series involves *disentangling global and local components* by decomposing the input tensor $\mathbf{X} \in \mathbb{R}^{B \times T \times D}$ into separate components that capture both time-invariant (global) and time-varying (local) structures. Concretely, one can write

$$\hat{\mathbf{X}} \;=\; \left(\mathbf{U}_{\text{fix}} + \Delta\mathbf{U}_t\right)\mathbf{V}$$

where $\mathbf{U}_{\text{fix}}$ is a shared representation across all time steps, $\Delta\mathbf{U}_{\text{t}}$ represents the per-time-step deviation, and $\mathbf{V}$ is a latent temporal representation. This decomposition naturally disentangles stable trends from more transient behaviors, enabling clearer interpretability and more focused regularization.

In practice, one can define a reconstruction term

$$\mathcal{L}_{\text{rec}} \;=\; \left\|\mathbf{X} \;-\; \hat{\mathbf{X}}\right\|_F^2$$

To discourage overfitting and ensure that $\Delta\mathbf{U}_t$ remains small unless necessary, a regularization penalty

$$\mathcal{L}_{\text{reg}} \;=\; \lambda_{\text{reg}} \left\|\Delta\mathbf{U}_{\text{t}}\right\|_F^2$$

is added, where $\|\cdot\|_F$ denotes the Frobenius norm. Additionally, if forecasting is part of the objective, a term

$$\mathcal{L}_{\text{forecast}} \;=\; \left\|\mathbf{Y} \;-\; \hat{\mathbf{Y}}\right\|^2$$

can be incorporated, where $\mathbf{Y}$ is the ground-truth future sequence and $\hat{\mathbf{Y}}$ is the model's prediction (i.e. $\hat{\mathbf{Y}} = f(\mathbf{U}_{fix}, \Delta\mathbf{U}_t, \mathbf{V})$ for some learnable function f) . The overall loss thus becomes:

$$L \;=\; \mathcal{L}_{\text{reconstruction}} \;+\; \mathcal{L}_{\text{reg}} \;+\; \alpha\,\mathcal{L}_{\text{forecast}},$$

with α controlling the trade-off between reconstruction accuracy and forecasting performance. By clearly separating $\mathbf{U}_{\text{fix}}$ from $\Delta\mathbf{U}_t$, this approach provides a natural way to capture global, time-invariant patterns alongside local variations. It also simplifies the interpretation of learned parameters and facilitates the application of domain-specific constraints or regularizers, ultimately leading to models that are both robust and more transparent in their handling of multivariate time series.

6.2.10 Aligning Large Language Models with Time Series Tasks

Finally, an especially forward-looking question is how to *align large language models (LLMs) with time series tasks.* Given that LLMs have excelled in learning from vast sequences of tokens, one might ask whether time series can be discretized or tokenized in a manner analogous to text. Achieving such an alignment would enable the direct application (or fine-tuning) of LLMs on time series data, leveraging their capacity for sequence modeling and potentially unlocking new capabilities in forecasting, anomaly detection, or other tasks. However, the fundamental differences between natural language data and numeric time series—particularly regarding continuity, scale, and correlation structures—must be carefully addressed. Investigating how best to transform continuous signals into discrete tokens, how to preserve crucial time dependencies, and how to incorporate domain-specific inductive biases remain open questions. Nonetheless, bridging the gap between LLMs and time series could unify two previously separate research trajectories, fostering cross-pollination of ideas and techniques.

6.2.11 Conclusion

In summary, the domain of multivariate time series analysis is brimming with exciting opportunities. As datasets grow larger and more complex, the importance of effective, scalable, and interpretable models becomes ever more critical. By building upon the foundations laid by shallow channel-attention transformers, SAM, and theoretically motivated regularization, we can aspire to create advanced architectures that strike an optimal balance between performance, interpretability, and computational cost. Whether by refining SAM to handle anisotropic sharpness, extending our theoretical frameworks to deep networks, or harnessing the power of dimension reduction and foundation models, there is a vast horizon of research waiting to be explored. The work presented thus far has offered a personal contribution to this evolving landscape, and it is our hope that future efforts will continue to push the boundaries of what is possible in representation learning dor multivariate time series.

References

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser & Illia Polosukhin (2017). "Attention is All you Need". In: 30. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett. URL: https://arxiv.org/pdf/1706.03762.

Box, George EP & Gwilym M Jenkins (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day.

Hochreiter, Sepp & Jürgen Schmidhuber (1997a). "Long Short-Term Memory". In: *Neural Computation*. Vol. 9. 8, pp. 1735–1780.

Zhou, Haoyi, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong & Wancai Zhang (2021). "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting". In: *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*. Vol. 35. AAAI Press, pp. 11106–11115.

Wu, Haixu, Yaochen Xie, Qi Tian, et al. (2021). "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting". In: *Advances in Neural Information Processing Systems* 34, pp. 22419–22430.

Lai, Guan, Weipeng Chang, Yiming Yang & Hanxiao Liu (2018). "Modeling long-and short-term temporal patterns with deep neural networks". In: *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104.

Caruana, Rich (1997). "Multitask learning". In: *Machine learning* 28.1, pp. 41–75.

Ansari, Abdul Fatir et al. (2024). *Chronos: Learning the Language of Time Series*. arXiv: 2403.07815 [cs.LG]. URL: https://arxiv.org/abs/2403.07815.

Goswami, Mononito, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li & Artur Dubrawski (2024). *MOMENT: A Family of Open Time-series Foundation Models*. arXiv: 2402.03885 [cs.LG]. URL: https://arxiv.org/abs/2402.03885.

Woo, Gerald, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese & Doyen Sahoo (2024a). *Unified Training of Universal Time Series Forecasting Transformers*. arXiv: 2402.02592 [cs.LG]. URL: https://arxiv.org/abs/2402.02592.

Jin, Ming et al. (2024). *Time-LLM: Time Series Forecasting by Reprogramming Large Language Models*. arXiv: 2310.01728 [cs.LG]. URL: https://arxiv.org/abs/2310.01728.

Zhou, Tian, PeiSong Niu, Xue Wang, Liang Sun & Rong Jin (2023). *One Fits All:Power General Time Series Analysis by Pretrained LM*. arXiv: 2302.11939 [cs.LG]. URL: https://arxiv.org/abs/2302.11939.

Das, Abhimanyu, Weihao Kong, Rajat Sen & Yichen Zhou (2024). *A decoder-only foundation model for time-series forecasting*. arXiv: 2310.10688 [cs.CL]. URL: https://arxiv.org/abs/2310.10688.

Gamboa, Janine C. Borges (2017). *Deep Learning for Time-Series Analysis*. arXiv: 1701.01887 [cs.LG].

Oreshkin, Boris N., Dmitri Carpov, Nicolas Chapados & Yoshua Bengio (2020). "N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting". In: *International Conference on Learning Representations (ICLR)*.

Zhou, Tian, Ziqing Ma, Qingsong Wen, Xue Wang, Lixin Sun & Rong Jin (2022). "FEDformer: Frequency Enhanced Decomposed Transformer for Long-Term Series Forecasting". In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 27268–27286.

Nie, Jiawei, Qingsong Wen, Xiyang Dai, Wei Zhang, Lixin Sun & Lingjuan Yang (2023). "A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers". In: *International Conference on Learning Representations (ICLR)*.

Woo, Wonkeun, B. Lim, S. Muni & B. Marr (2022). "ETSformer: Exponential Smoothing Transformers for Time-Series Forecasting". In: *arXiv preprint arXiv:2202.01381*.

Zhang, Yue, Qin Liu, Shu Wu & Liang Wang (2021). "Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Yun, Chulhee, Srinadh Bhojanapalli, Aditya S. Rawat, Sashank J. Reddi & Sanjiv Kumar (2019). "Are Transformers Universal Approximators of Sequence-to-Sequence Functions?" In: *Advances in Neural Information Processing Systems*. Vol. 32.

Bengio, Yoshua, Patrice Simard & Paolo Frasconi (1994). "Learning Long-Term Dependencies with Gradient Descent is Difficult". In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166.

Glorot, Xavier & Yoshua Bengio (2010a). "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR Workshop and Conference Proceedings, pp. 249–256.

Wen, Qingsong, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan & Liang Sun (2023). *Transformers in Time Series: A Survey*. arXiv: 2202.07125 [cs.LG]. URL: https://arxiv.org/abs/2202.07125.

Zeng, Ailing, Muxi Chen, Lei Zhang & Qiang Xu (2023). "Are Transformers Effective for Time Series Forecasting?" In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Chen, Xiangning, Cho-Jui Hsieh & Boqing Gong (2022). "When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=LtKcMgGOeLt.

Zhai, Shuangfei, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu & Joshua M. Susskind (July 2023). "Stabilizing Transformer Training by Preventing Attention Entropy Collapse". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato & Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 40770–40803. URL: https://proceedings.mlr.press/v202/zhai23a.html.

Feofanov, Vasilii, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang & Ievgen Redko (2024). "Mantis: A Foundation Model for Time Series Classification". In: *arXiv preprint arXiv:2502.15637*. URL: https://arxiv.org/abs/2502.15637.

Hamilton, James D. (1994). *Time Series Analysis*. Princeton University Press.

Brockwell, Peter J. & Richard A. Davis (2009). *Time Series: Theory and Methods*. 2nd. Springer.

Tsay, Ruey S. (2013). *Multivariate Time Series Analysis: With R and Financial Applications*. John Wiley & Sons.

Cleveland, Robert B, William S Cleveland, Jean E McRae & Irma Terpenning (1990). "STL: A Seasonal-Trend Decomposition Procedure Based on Loess". In: *Journal of Official Statistics* 6.1, pp. 3–73.

Little, Roderick JA & Donald B Rubin (2002). *Statistical Analysis with Missing Data*. John Wiley & Sons.

Chandola, Varun, Arindam Banerjee & Vipin Kumar (2009). "Anomaly Detection: A Survey". In: *ACM Computing Surveys* 41.3, pp. 1–58.

Boniol, Paul, Qinghua Liu, Mingyi Huang, Themis Palpanas & John Paparrizos (2024). *Dive into Time-Series Anomaly Detection: A Decade Review*. arXiv: 2412.20512 [cs.LG]. URL: https://arxiv.org/abs/2412.20512.

Friedman, Jerome H (1984). "A Variable Span Smoother". In: *Laboratory for Computational Statistics, Stanford University, Technical Report No. 5*.

Ye, Lexiang & Eamonn Keogh (2009). "Time Series Shapelets: A New Primitive for Data Mining". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 947–956.

Hills, Jon, Jason Lines, Eugen Baranauskas, James Mapp & Anthony Bagnall (2014). "Classification of Time Series by Shapelet Transformation". In: *Data Mining and Knowledge Discovery* 28.4, pp. 851–881.

Xuan, Xiang & Kevin Murphy (2007). "Modeling changing dependency structure in multivariate time series". In: *Proceedings of the 24th international conference on Machine learning*, pp. 1055–1062.

Zhao, Lifan & Yanyan Shen (2024). “Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators”. In: *International Conference on Learning Representations (ICLR)*.

Wu, Zonghan, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang & Chengqi Zhang (2020). *Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks*. arXiv: 2005.11650 [cs.LG]. URL: https://arxiv.org/abs/2005.11650.

Ismail Fawaz, Hassan, Germain Forestier, Jonathan Weber, Lhassane Idoumghar & Pierre-Alain Muller (2019). “Deep learning for time series classification: a review”. In: *Data Mining and Knowledge Discovery* 33, pp. 917–963.

Bagnall, Anthony, Jason Lines, Aaron Bostrom, James Large & Eamonn Keogh (2017). “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances”. In: *Data Mining and Knowledge Discovery* 31, pp. 606–660.

Baydogan, Mustafa G & George Runger (2016). “Time series representation and similarity based on local autopatterns”. In: *Data Mining and Knowledge Discovery* 30, pp. 476–509.

Ding, Hui, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang & Eamonn Keogh (2008). “Querying and mining of time series data: Experimental comparison of representations and distance measures”. In: *Proceedings of the VLDB Endowment* 1.2, pp. 1542–1552.

Shokoohi-Yekta, Mohammad, Bei Hu, Yanping Jin, Jun Wang & Eamonn Keogh (2017). “Generalizing dynamic time warping to the multi-dimensional case requires an adaptive approach”. In: *Data Mining and Knowledge Discovery* 31, pp. 1–31.

Kate, Rohit J (2016). “Using dynamic time warping distances as features for improved time series classification”. In: *Data Mining and Knowledge Discovery* 30.2, pp. 283–312.

Berndt, Donald J & James Clifford (1994). “Using dynamic time warping to find patterns in time series”. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 359–370.

Rakthanmanon, Thanawin et al. (2012). “Searching and mining trillions of time series subsequences under dynamic time warping”. In: *ACM Transactions on Knowledge Discovery from Data* 7.3, pp. 1–31.

Sakoe, Hiroaki & Seibi Chiba (1978). “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing*. Vol. 26. 1, pp. 43–49.

Itakura, Fumitada (1975). “Minimum prediction residual principle applied to speech recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.1, pp. 67–72. DOI: 10.1109/TASSP.1975.1162641.

Keogh, Eamonn & Chotirat Ann Ratanamahatana (2005). “Exact indexing of dynamic time warping”. In: *Knowledge and Information Systems* 7.3, pp. 358–386.

Salvador, Stan & Philip Chan (2007a). “Toward accurate dynamic time warping in linear time and space”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 560–567.

Cuturi, Marco & Mathieu Blondel (2017). “Soft-DTW: a differentiable loss function for time-series”. In: *International Conference on Machine Learning (ICML)*, pp. 894–903.

Chen, Lei, M. Tamer Özsu & Vincent Oria (2005). "Robust and fast similarity search for moving object trajectories". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, pp. 491–502.

Vlachos, Michail, Michalis Hadjieleftheriou, Eamonn Keogh & Dimitrios Gunopulos (2002). "Indexing multi-dimensional time-series with support for multiple distance measures". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 216–225.

Marteau, Pierre-François (2009). "Time warp edit distance with stiffness adjustment for time series matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2, pp. 306–318.

Dau, Hoang Anh, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Sylvain Aregui, Nurjahan Begum, Abdullah Mueen & Eamonn Keogh (2019). "The UCR Time Series Archive". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 2231–2240.

Lines, Jason, L. M. Davis, Jon Hills & Anthony Bagnall (2015). "A shapelet transform for time series classification". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 289–298.

Fulcher, Ben D. & Nick S. Jones (2014). "Highly comparative feature-based time-series classification". In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 3026–3037.

Deng, H., G. Runger, E. Tuv & M. Vladimir (2013). "A time series forest for classification and feature extraction". In: *Information Sciences* 239, pp. 142–153.

Hjorth, Bo (1970). "EEG analysis based on time domain properties". In: *Electroencephalography and Clinical Neurophysiology* 29.3, pp. 306–310.

Keogh, Eamonn & Shruti Kasetty (2001). "On the need for time series data mining benchmarks: A survey and empirical demonstration". In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 102–111.

Addison, Paul S (2017). *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*. CRC Press.

Huang, Norden E, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung & Henry H Liu (1998). "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.197.

Karlsson, Ingmar, Panagiotis Papapetrou, Antti Ukkonen, Gareth Tyson & Martti Juhola (2016). "Generalized Random Shapelet Forest". In: *2016 IEEE International Conference on Data Mining (ICDM)*, pp. 1–10.

Lin, Jessica, Eamonn Keogh, Stefano Lonardi & Bill Chiu (2003). "A symbolic representation of time series, with implications for streaming algorithms". In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 2–11.

Schäfer, Patrick (2015). “The BOSS is concerned with time series classification in the presence of noise”. In: *Data Mining and Knowledge Discovery* 29.6, pp. 1505–1530.

Senin, Pavel & Sergey Malinchik (2013). “SAX-VSM: Interpretable time series classification using SAX and vector space model”. In: *2013 IEEE 13th International Conference on Data Mining*, pp. 1175–1180.

Lines, Jason & Anthony Bagnall (2018). “Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles”. In: *ACM Transactions on Knowledge Discovery from Data* 12.5, pp. 1–35.

Lines, Jason, Sarah Taylor & Anthony Bagnall (July 2018). “Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles”. In: *ACM Trans. Knowl. Discov. Data* 12.5. ISSN: 1556-4681. DOI: 10.1145/3182382. URL: https://doi.org/10.1145/3182382.

Middlehurst, Matthew, James Large, Michael Flynn, Jason Lines, Aaron Bostrom & Anthony J. Bagnall (2021). “HIVE-COTE 2.0: a new meta ensemble for time series classification”. In: *CoRR* abs/2104.07551. arXiv: 2104.07551. URL: https://arxiv.org/abs/2104.07551.

Wang, Zhiguang, Weizhong Yan & Tim Oates (2017). “Time series classification from scratch with deep neural networks: A strong baseline”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren & Jian Sun (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778.

Ismail Fawaz, Hassan, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar & Pierre Muller (2020). “InceptionTime: Finding AlexNet for time series classification”. In: *Data Mining and Knowledge Discovery*. Vol. 34. 6, pp. 1936–1962.

Liu, Minghao, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang & Wei Song (2021). “Gated Transformer Networks for Multivariate Time Series Classification”. In: *arXiv preprint arXiv:2103.14438*. URL: https://arxiv.org/abs/2103.14438.

Bagnall, Anthony, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam & Eamonn Keogh (2018a). “The UEA multivariate time series classification archive, 2018”. In: *arXiv preprint arXiv:1811.00075*.

Goldberger, Ary L., Luís A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, C.-K. Peng & H. Eugene Stanley (2000). “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals”. In: *Circulation* 101.23, e215–e220.

Hložek, R. et al. (2020). *Results of the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC)*. arXiv: 2012.12392 [astro-ph.IM]. URL: https://arxiv.org/abs/2012.12392.

Fawcett, Tom (2006). “An introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8, pp. 861–874.

Saito, Takaya & Marc Rehmsmeier (2015). "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS one* 10.3, e0118432.

Dempster, Angus, François Petitjean & Geoffrey I. Webb (2020a). "ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels". In: *Data Mining and Knowledge Discovery* 34.5, pp. 1454–1495.

Tan, Chang Wei, Angus Dempster, Christoph Bergmeir & Geoffrey I. Webb (2022). *MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification*. arXiv: 2102.00457 [cs.LG]. URL: https://arxiv.org/abs/2102.00457.

Shifaz, Abdullah, Charlotte Pelletier, Francois Petitjean & Geoffrey I. Webb (2020). "TS-CHIEF: A scalable and accurate forest algorithm for time series classification". In: *Data Mining and Knowledge Discovery* 34.3, pp. 742–775.

Schäfer, Patrick & Ulf Leser (2017). "Multivariate Time Series Classification with WEASEL+MUSE". In: *arXiv preprint arXiv:1711.11343*.

Zerveas, George, Shashank Jayaraman, Deepak B. Patel, Ashok Bhamidipaty & Carsten Eickhoff (2021). "A transformer-based framework for multivariate time series representation learning". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124.

Nie, Yixuan, Nam H Nguyen, Phanwadee Sinthong & Jayant Kalagnanam (2023). "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis". In: *arXiv preprint arXiv:2302.04615*.

Karim, Fazle, Somshubra Majumdar, Homa Darabi & Samuel Harford (2019). "Multivariate lstm-fcns for time series classification". In: *Neural Networks* 116, pp. 237–245.

Seyfi, Ali, Jean-Francois Rajotte & Raymond T. Ng (2022). "Generating multivariate time series with COmmon Source CoordInated GAN (COSCI-GAN)". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave & Kyunghyun Cho. URL: https://openreview.net/forum?id=RP1CtZhEmR.

Box, George E. P., Gwilym M. Jenkins, Gregory C. Reinsel & Greta M. Ljung (2015). *Time Series Analysis: Forecasting and Control*. 5th. Wiley.

Hyndman, Rob J. & George Athanasopoulos (2008). *Forecasting: Principles and Practice*. https://otexts.com/fpp2/. OTexts.

Gardner, Everette S. (2006). "Exponential smoothing: The state of the art—Part II". In: *International Journal of Forecasting* 22.4, pp. 637–666.

Hyndman, Rob J., Anne B. Koehler, J. Keith Ord & Ralph D. Snyder (2002). "A state space framework for automatic forecasting using exponential smoothing methods". In: *International Journal of Forecasting* 18.3, pp. 439–454.

Livera, A. M. De, Rob J. Hyndman & Ralph D. Snyder (2011). "Forecasting time series with complex seasonal patterns using exponential smoothing". In: *Journal of the American Statistical Association* 106.496, pp. 1513–1527.

Taylor, Sean J. & Benjamin Letham (2018). "Forecasting at scale". In: *The American Statistician* 72.1, pp. 37–45.

Ahmed, Nasrin M., Amir Atiya, Neamat El Gayar & Hisham El-Shishiny (2010). "An Empirical Comparison of Machine Learning Models for Time Series Forecasting". In: *Econometric Reviews* 29.5-6, pp. 594–621.

Bontempi, Gianluca, Souhaib Ben Taieb & Yann-Aël Le Borgne (2012). "Machine Learning Strategies for Time Series Forecasting". In: *European Business Intelligence Summer School*, pp. 62–77.

Breiman, Leo (2001). "Random forests". In: *Machine Learning* 45.1, pp. 5–32.

Chen, Tianqi & Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye & Tie-Yan Liu (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3146–3154.

Dorogush, Anna Veronika, Vasily Ershov & Andrey Gulin (2018). "CatBoost: Gradient Boosting with Categorical Features Support". In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6639–6649.

Smola, Alex J. & Bernhard Schölkopf (2004). "A tutorial on support vector regression". In: *Statistics and Computing* 14, pp. 199–222.

Zhang, G., B. Eddy Patuwo & M. Y. Hu (1998). "Forecasting with artificial neural networks: The state of the art". In: *International Journal of Forecasting* 14.1, pp. 35–62.

Harvey, Andrew C. (1993). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

Wolpert, David H. (1992). "Stacked Generalization". In: *Neural Networks* 5.2, pp. 241–259.

Makridakis, Spyros, Evangelos Spiliotis & Vassilios Assimakopoulos (2020). "The M5 Accuracy Competition: Results, Findings, and Conclusions". In: *International Journal of Forecasting* 37, pp. 199–230.

Zhang, G. (2003). "Time series forecasting using a hybrid ARIMA and neural network model". In: *Neurocomputing* 50, pp. 159–175.

Casolaro, Angelo, Vincenzo Capone, Gennaro Iannuzzo & Francesco Camastra (2023). "Deep Learning for Time Series Forecasting: Advances and Open Problems". In: *Information* 14.11. ISSN: 2078-2489. DOI: 10.3390/info14110598. URL: https://www.mdpi.com/2078-2489/14/11/598.

Hochreiter, Sepp & Jürgen Schmidhuber (1997b). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk & Yoshua Bengio (Oct. 2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang & Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://aclanthology.org/D14-1179.

Salinas, David, Valentin Flunkert, Jan Gasthaus & Tim Januschowski (2020a). "DeepAR: Probabilistic forecasting with autoregressive recurrent networks". In: *International Journal of Forecasting* 36.3, pp. 1181–1191.

Bai, Shaojie, J. Zico Kolter & Vladlen Koltun (2018). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271*.

Oord, Aaron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior & Koray Kavukcuoglu (2016). "WaveNet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499*.

Lai, Guokun, Wei-Cheng Chang, Yiming Yang & Hanxiao Liu (2018a). "Modeling long- and short-term temporal patterns with deep neural networks". In: *ACM SIGIR Conference on Research and Development in Information Retrieval*.

Wu, Zonghan, Shirui Pan, Guodong Long, Jing Jiang, Chengqi Zhang & Philip S. Yu (2020). "Connecting the dots: Multivariate time series forecasting with graph neural networks". In: *arXiv preprint arXiv:2005.11650*.

Nie, Yuqing et al. (2023). "A time series is worth 64 words: Long-term forecasting with transformers". In: *ICLR*.

Liu, Yong, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma & Mingsheng Long (2024). "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=JePfAI8fah.

Kitaev, Nikita, Lukasz Kaiser & Anselm Levskaya (2020). "Reformer: The Efficient Transformer". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rkgNKkHtvB.

Li, Shiyang, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang & Xifeng Yan (2019). "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox & R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/6775a0635c302542da2c32aa19d86be0-Paper.pdf.

Zeng, Ailing, Junchi Yan, Ya Zhang, Kai Zheng, Feng Xu & Xuanheng Xu (2023). "Are Transformers Effective for Time Series Forecasting?" In: *AAAI Conference on Artificial Intelligence*.

Dong, Yihe, Jean-Baptiste Cordonnier & Andreas Loukas (July 2021). "Attention is not all you need: pure attention loses rank doubly exponentially with depth". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila & Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 2793–2803. URL: https://proceedings.mlr.press/v139/dong21a.html.

Liu, Liyuan, Xiaodong Liu, Jianfeng Gao, Weizhu Chen & Jiawei Han (2020). "Understanding the Difficulty of Training Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Foret, Pierre, Ariel Kleiner, Hossein Mobahi & Behnam Neyshabur (2021). "Sharpness-aware Minimization for Efficiently Improving Generalization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=6Tm1mposlrM.

Tolstikhin, Ilya et al. (2021). *MLP-Mixer: An all-MLP Architecture for Vision*. arXiv: 2105.01601 [cs.CV].

Liu, Yujie, Xiyuan Li & Yipeng Wang (2023). "TSMixer: A MLP-Mixer Architecture for Time Series Forecasting". In: *ICLR*.

Makridakis, Spyros & Michele Hibon (2000). "The M3-Competition: Results, Conclusions and Implications". In: *International Journal of Forecasting* 16.4, pp. 451–476.

Makridakis, Spyros, Evangelos Spiliotis & Vassilios Assimakopoulos (2018). "The M4 Competition: Results, findings, conclusion and way forward". In: *International Journal of Forecasting* 34.4, pp. 802–808.

Godahewa, Ruwan, Christoph Bergmeir, Geoffrey I Webb, Carl Lubba & Ben D Fulcher (2021). "Monash Time Series Forecasting Archive". In: *arXiv preprint arXiv:2105.06643*.

Athanasopoulos, George, Rob J Hyndman, Haiyan Song & Debbie C Wu (2011). "The Tourism Forecasting Competition". In: *International Journal of Forecasting* 27.3, pp. 822–844.

Yu, Rose, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek & Yan Liu (June 2017). "Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting". In: pp. 777–785. ISBN: 978-1-61197-497-3. DOI: 10.1137/1.9781611974973.87.

Dua, Dheeru & Casey Graff (2017). "UCI Machine Learning Repository". In.

Google (2017). *Kaggle Web Traffic Time Series Forecasting Competition*. Available at: https://www.kaggle.com/c/web-traffic-time-series-forecasting.

Zeng, Ailing, Shiyang Li, Yao Liu, Junchi Zhang, Jingkang Zhou & Li Jiang (2023). "Are Transformers Effective for Time Series Forecasting?" In: *NeurIPS*.

Nie, Yuqi, Nam H Nguyen, Phanwadee Sinthong & Jayant Kalagnanam (2022). "A time series is worth 64 words: Long-term forecasting with transformers". In: *arXiv preprint arXiv:2211.14730*.

Challu, Christopher, Grzegorz Marcjasz, Rafal Weron, David Salinas & Jan Gasthaus (2022). "N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting". In: *NeurIPS*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren & Jian Sun (2015). "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Dosovitskiy, Alexey et al. (2021a). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *International Conference on Learning Representations (ICLR)*.

Achiam, Joshua, Shahar Adler, Sharan Agarwal, Latifa Ahmad, Ilge Akkaya, Fernando Luis Aleman, Daniel Almeida, Johannes Altenschmidt, Sam Altman, et al. (2023). "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774*.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023a). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Sanjeev Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. (2021). "On the Opportunities and Risks of Foundation Models". In: *arXiv preprint arXiv:2108.07258*.

Rasul, Kashif, Anikait Ashok, Andy R. Williams, et al. (2023). "Lag-LLAMA: Towards foundation models for time series forecasting". In: *R0-FoMo Workshop, NeurIPS*.

Woo, Gerald, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese & Doyen Sahoo (2024b). "Unified training of universal time series forecasting transformers". In: *International Conference on Machine Learning (ICML)*.

Wang, Yipeng, Yixuan Qiu, Peng Chen, Kai Zhao, Yu Shu, Ziang Rao, Lujia Pan, Bo Yang & Chenghong Guo (2024). "ROSE: Register assisted general time series forecasting with decomposed frequency learning". In: *arXiv preprint arXiv:2405.17478*.

Yue, Zhen, Ying Wang, Jian Duan, Tao Yang, Cheng Huang, Yifan Tong & Bo Xu (2022). "TS2Vec: Towards Universal Representation of Time Series". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 8, pp. 8980–8987.

Tonekaboni, Saeed et al. (2021). "Temporal Neighborhood Coding for Unsupervised Time Series Representation Learning". In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*.

Trindade, A. (2015). *Electricity Dataset*. https://archive.org/....

California Department of Transportation (2024). *Traffic Dataset*. https://dot.ca.gov/....

Max Planck Institute for Biogeochemistry (2024). *Weather Dataset*. https://www.bgc-jena.mpg.de/....

Centers for Disease Control and Prevention (2024). *ILI Dataset*. https://www.cdc.gov/....

Lai, Guokun, Wei-Cheng Chang, Yiming Yang & Hanxiao Liu (2018b). "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks". In: *SIGIR*.

Wu, X., Y. Wang & Z. Zhao (2023). "Title of the Paper". In: *Conference/Journal*.

Nie, L., Q. Li & H. Zhou (2023). "Title of the Paper". In: *Conference/Journal*.

Challu, C., R. Egri & S. Gagnon (2023). "Title of the Paper". In: *Conference/Journal*.

Godahewa, R., P. Montero-Manso & R. Hyndman (2021). *Title of the Paper*. Monash Time Series Forecasting Archive.

Paparrizos, J., S. Wu & J. Subramanian (2022). *Title of the Paper*. TSB-UAD Anomaly Benchmark.

Pan, Sinno Jialin & Qiang Yang (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359.

Čepulionis, Paulius & Kristina Lukoševičiūtė (June 2016). "Electrocardiogram time series forecasting and optimization using ant colony optimization algorithm". In: *Mathematical Models in Engineering* 2.1, pp. 69–77. ISSN: 2351-5279. URL: https://www.extrica.com/article/17229.

UCI (2015). *Electricity dataset*. URL: https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams2011201 (visited on 01/13/2024).

Max Planck Institute (2021). *Weather dataset*. URL: https://www.bgc-jena.mpg.de/wetter/ (visited on 01/13/2024).

Sonkavde, Gaurang, Deepak Sudhakar Dharrao, Anupkumar M. Bongale, Sarika T. Deokate, Deepak Doreswamy & Subraya Krishna Bhat (2023). "Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications". In: *International Journal of Financial Studies* 11.3. ISSN: 2227-7072. DOI: 10.3390/ijfs11030094. URL: https://www.mdpi.com/2227-7072/11/3/94.

Sorjamaa, Antti, Jin Hao, Nima Reyhani, Yongnan Ji & Amaury Lendasse (2007). "Methodology for long-term prediction of time series". In: *Neurocomputing* 70.16. Neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN 2005), pp. 2861–2869. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2006.06.015. URL: https://www.sciencedirect.com/science/article/pii/S0925231207001610.

Chen, Renyi & Molei Tao (July 2021). "Data-driven Prediction of General Hamiltonian Dynamics via Learning Exactly-Symplectic Maps". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila & Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 1717–1727. URL: https://proceedings.mlr.press/v139/chen21r.html.

Box, George Edward Pelham & Gwilym Jenkins (1990). *Time Series Analysis, Forecasting and Control*. USA: Holden-Day, Inc. ISBN: 0816211043.

Box, G. E. P., G. M. Jenkins & J. F. MacGregor (June 1974). "Some Recent Advances in Forecasting and Control". In: *Journal of the Royal Statistical Society Series C* 23.2, pp. 158–179. DOI: 10.2307/2346997. URL: https://ideas.repec.org/a/bla/jorssc/v23y1974i2p158-179.html.

Rangapuram, Syama Sundar, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang & Tim Januschowski (2018). "Deep State Space Models for Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi & R. Garnett. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf.

Salinas, David, Valentin Flunkert, Jan Gasthaus & Tim Januschowski (2020b). "DeepAR: Probabilistic forecasting with autoregressive recurrent networks". In: *International Journal of Forecasting* 36.3, pp. 1181–1191. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2019.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301888.

Fan, Chenyou, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei & Heng Huang (2019). "Multi-Horizon Time Series Forecasting with Temporal Attention Learning". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: As-

sociation for Computing Machinery, pp. 2527–2535. ISBN: 9781450362016. DOI: 10.1145/3292500.3330662. URL: https://doi.org/10.1145/3292500.3330662.

Lai, Guokun, Wei-Cheng Chang, Yiming Yang & Hanxiao Liu (2018c). "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: Association for Computing Machinery, pp. 95–104. ISBN: 9781450356572. DOI: 10.1145/3209978.3210006. URL: https://doi.org/10.1145/3209978.3210006.

Sen, Rajat, Hsiang-Fu Yu & Inderjit Dhillon (2019). "Think globally, act locally: a deep neural network approach to high-dimensional time series forecasting". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. URL: http://arxiv.org/abs/1810.04805.

Radford, Alec, Karthik Narasimhan, Tim Salimans & Ilya Sutskever (2018). "Improving language understanding by generative pre-training". In: *2018 OpenAI Tech Report*.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023b). *LLaMA: Open and Efficient Foundation Language Models*. cite arxiv:2302.13971. URL: http://arxiv.org/abs/2302.13971.

OpenAI (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].

Dosovitskiy, Alexey et al. (2021b). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=YicbFdNTTy.

Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski & Armand Joulin (2021). "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Touvron, Hugo, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles & Herve Jegou (July 2021). "Training data-efficient image transformers and distillation through attention". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila & Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 10347–10357. URL: https://proceedings.mlr.press/v139/touvron21a.html.

Nie, Yuqi, Nam H Nguyen, Phanwadee Sinthong & Jayant Kalagnanam (2023). "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Jbdc0vTOcol.

Chen, Si-An, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder & Tomas Pfister (2023). "TSMixer: An All-MLP Architecture for Time Series Forecasting". In: *Transactions*

on Machine Learning Research. ISSN: 2835-8856. URL: https://openreview.net/forum?id=wbpxTuXgm0.

Liu, Shizhan, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu & Schahram Dustdar (2022). "Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=0EXmFzUn5I.

Cirstea, Razvan-Gabriel, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong & Shirui Pan (July 2022). "Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, pp. 1994–2001. DOI: 10.24963/ijcai.2022/277. URL: https://doi.org/10.24963/ijcai.2022/277.

Kim, Taesung, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi & Jaegul Choo (2021). "Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=cGDAkQo1C0p.

Zhang, Hang et al. (June 2022). "ResNeSt: Split-Attention Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2736–2746.

Zamir, Syed Waqas, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan & Ming-Hsuan Yang (2022). "Restormer: Efficient Transformer for High-Resolution Image Restoration". In: *CVPR*.

Glorot, Xavier & Yoshua Bengio (May 2010b). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh & Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

Daneshmand, Hadi, Jonas Kohler, Francis Bach, Thomas Hofmann & Aurelien Lucchi (2020). "Batch normalization provably avoids rank collapse for randomly initialised deep networks". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc. ISBN: 9781713829546.

Kim, Hyunjik, George Papamakarios & Andriy Mnih (July 2021). "The Lipschitz Constant of Self-Attention". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila & Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 5562–5571. URL: https://proceedings.mlr.press/v139/kim21i.html.

Recht, Benjamin, Maryam Fazel & Pablo A. Parrilo (2010). "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization". In: *SIAM Review* 52.3, pp. 471–501. DOI: 10.1137/070697835. eprint: https://doi.org/10.1137/070697835. URL: https://doi.org/10.1137/070697835.

Recht, Benjamin (Dec. 2011). "A Simpler Approach to Matrix Completion". In: *J. Mach. Learn. Res.* 12.null, pp. 3413–3430. ISSN: 1532-4435.

Candès, Emmanuel & Benjamin Recht (June 2012). "Exact matrix completion via convex optimization". In: *Commun. ACM* 55.6, pp. 111–119. ISSN: 0001-0782. DOI: 10.1145/2184319.2184343. URL: https://doi.org/10.1145/2184319.2184343.

Anagnostidis, Sotiris, Luca Biggio, Lorenzo Noci, Antonio Orvieto, Sidak Pal Singh & Aurelien Lucchi (2022). "Signal Propagation in Transformers: Theoretical Perspectives and the Role of Rank Collapse". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave & Kyunghyun Cho. URL: https://openreview.net/forum?id=FxVH7iToXS.

Kingma, Diederik & Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA.

Loshchilov, Ilya & Frank Hutter (2017). "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Skq89Scxx.

California Department of Transportation (2021). *Traffic dataset*. URL: https://pems.dot.ca.gov/ (visited on 01/13/2024).

He, Bobby, James Martens, Guodong Zhang, Aleksandar Botev, Andrew Brock, Samuel L Smith & Yee Whye Teh (2023). "Deep Transformers without Shortcuts: Modifying Self-attention for Faithful Signal Propagation". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=NPrsUQgMjKK.

Trockman, Asher & J. Zico Kolter (2023). "Mimetic Initialization of Self-Attention Layers". In: *Proceedings of the 40th International Conference on Machine Learning*. ICML'23. Honolulu, Hawaii, USA: JMLR.org.

Woo, Gerald, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese & Doyen Sahoo (2024c). *Unified Training of Universal Time Series Forecasting Transformers*. arXiv: 2402.02592 [cs.LG].

Ahn, Kwangjun, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie & Suvrit Sra (2023). *Linear attention is (maybe) all you need (to understand transformer optimization)*. arXiv: 2310.01082 [cs.LG].

Pan, Yan & Yuanzhi Li (2022). "Toward Understanding Why Adam Converges Faster Than SGD for Transformers". In: *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*. URL: https://openreview.net/forum?id=Sf1NlV2r6PO.

Zhang, Jingzhao, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar & Suvrit Sra (2020). "Why are Adaptive Methods Good for Attention Models?" In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan & H. Lin. Vol. 33. Curran Associates, Inc., pp. 15383–15393. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/b05b57f6add810d3b7490866d74Paper.pdf.

Nesterov, Yurii (1983). "A method for solving the convex programming problem with convergence rate $O(1/k^2)$". In: *Proceedings of the USSR Academy of Sciences* 269, pp. 543–547. URL: https://api.semanticscholar.org/CorpusID:145918791.

Loshchilov, Ilya & Frank Hutter (2019). “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Shao, Ling (2015). “Transfer learning for visual categorization: a survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 26.5, pp. 1019–1034. DOI: 10.1109/TNNLS.2015.2390198.

Ruder, Sebastian, Matthew E. Peters, Swabha Swayamdipta & Thomas Wolf (2019). “Evolution of Transfer Learning in Natural Language Processing”. In: *arXiv preprint arXiv:1910.07370*.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li & Peter J. Liu (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21, pp. 1–67.

Mei, Suyu, Wang Fei & Shuigeng Zhou (2011). “Gene ontology based transfer learning for protein subcellular localization”. In: *BMC bioinformatics* 12, pp. 1–12.

Shin, Hoo-Chang, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura & Ronald M Summers (2016). “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning”. In: *IEEE transactions on medical imaging* 35.5, pp. 1285–1298.

Hu, Yiming, Mo Li, Qiongshi Lu, Haoyi Weng, Jiawei Wang, Seyedeh M Zekavat, Zhaolong Yu, Boyang Li, Jianlei Gu, Sydney Muchnik, et al. (2019). “A statistical framework for cross-tissue transcriptome-wide association analysis”. In: *Nature genetics* 51.3, pp. 568–576.

Bai, Zhidong & Jack W Silverstein (2010). *Spectral Analysis of Large Dimensional Random Matrices*. Springer Series in Statistics. Springer New York, NY. ISBN: 978-1-4419-0660-1. DOI: 10.1007/978-1-4419-0661-8.

Tao, Terence (2012). *Topics in Random Matrix Theory*. Vol. 132. Graduate Studies in Mathematics. American Mathematical Society. ISBN: 978-0821885079.

Yang, Fan, Hongyang R. Zhang, Sen Wu, Christopher Ré & Weijie J. Su (2023). *Precise High-Dimensional Asymptotics for Quantifying Heterogeneous Transfers*. arXiv: 2010.11750 [stat.ML].

Li, Sai, T Tony Cai & Hongzhe Li (2022). “Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84.1, pp. 149–173.

Mousavi Kalan, Mohammadreza, Zalan Fabian, Salman Avestimehr & Mahdi Soltanolkotabi (2020). “Minimax lower bounds for transfer learning with linear and one-hidden layer neural networks”. In: *Advances in Neural Information Processing Systems* 33, pp. 1959–1969.

Nguyen, Minh-Toan & Romain Couillet (2023). “Asymptotic Bayes risk of semi-supervised multitask learning on Gaussian mixture”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 5063–5078.

Ilbert, Romain, Ambroise Odonnat, Vasilii Feofanov, Aladin Virmaux, Giuseppe Paolo, Themis Palpanas & Ievgen Redko (2024). *Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention*. arXiv: 2402.10198 [cs.LG].

Wainwright, Martin J (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press.

Couillet, Romain & Zhenyu Liao (2022). *Random matrix methods for machine learning*. Cambridge University Press.

Feofanov, Vasilii, Malik Tiomoko & Aladin Virmaux (July 2023). "Random Matrix Analysis to Balance between Supervised and Unsupervised Learning under the Low Density Separation Assumption". In: *Proceedings of the 40th International Conference on Machine Learning*, pp. 10008–10033. URL: https://proceedings.mlr.press/v202/feofanov23a.html.

Seddik, Mohamed El Amine, Cosme Louart, Mohamed Tamaazousti & Romain Couillet (2020). "Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures". In: *International Conference on Machine Learning*.

Couillet, Romain & Merouane Debbah (2011). *Random matrix methods for wireless communications*. Cambridge University Press.

Potters, M., J.-P. Bouchaud & L. Laloux (2005). "Financial applications of random matrix theory: Old laces and new pieces". In: *Acta Physica Polonica B* 36.9, p. 2767.

Tiomoko, Malik, Romain Couillet & Hafiz Tiomoko (2020). *Large Dimensional Analysis and Improvement of Multi Task Learning*. arXiv: 2009.01591 [stat.ML].

Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. (2023). "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*.

Wang, Yihang, Yuying Qiu, Peng Chen, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang & Chenjuan Guo (2024). "ROSE: Register Assisted General Time Series Forecasting with Decomposed Frequency Learning". In: *arXiv preprint arXiv:2405.17478*.

Garza, Azul & Max Mergenthaler-Canseco (2023). "TimeGPT-1". In: *arXiv preprint arXiv:2310.03589*.

Lin, Chenguo, Xumeng Wen, Wei Cao, Congrui Huang, Jiang Bian, Stephen Lin & Zhirong Wu (2024). "NuTime: Numerically Multi-Scaled Embedding for Large- Scale Time-Series Pretraining". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=TwiSBZ0p9u.

Salvador, Stan & Philip Chan (2007b). "Toward accurate dynamic time warping in linear time and space". In: *Intelligent Data Analysis* 11.5, pp. 561–580.

Lines, Jason, Luke M Davis, Jon Hills & Anthony Bagnall (2012). "A shapelet transform for time series classification". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 289–297.

Deng, Houtao, George Runger, Eugene Tuv & Vladimir Martyanov (2013). "A time series forest for classification and feature extraction". In: *Information Sciences* 239, pp. 142–153.

Lin, Jessica, Eamonn Keogh, Li Wei & Stefano Lonardi (2007). "Experiencing SAX: a novel symbolic representation of time series". In: *Data Mining and knowledge discovery* 15, pp. 107–144.

Lin, Jessica, Rohan Khade & Yuan Li (2012). "Rotation-invariant similarity in time series using bag-of-patterns representation". In: *Journal of Intelligent Information Systems* 39, pp. 287–315.

Dempster, Angus, François Petitjean & Geoffrey I. Webb (July 2020b). "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels". In: *Data Mining and Knowledge Discovery* 34.5, pp. 1454–1495. ISSN: 1573-756X. DOI: 10.1007/s10618-020-00701-z. URL: http://dx.doi.org/10.1007/s10618-020-00701-z.

Rasul, Kashif, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, et al. (2023). "Lag-llama: Towards foundation models for time series forecasting". In: *arXiv preprint arXiv:2310.08278*.

Feofanov, Vasilii, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang & Ievgen Redko (2025). *Mantis: Foundation Model with Adapters for Multichannel Time Series Classification*. URL: https://github.com/vfeofanov/mantis.

Zhou, Tian, Peisong Niu, Xue Wang, Liang Sun & Rong Jin (2023). "One Fits All: Power General Time Series Analysis by Pretrained LM". In: *arXiv preprint arXiv:2302.11939*.

Wei, William WS (2018). *Multivariate time series analysis and applications*. John Wiley & Sons.

Bagnall, Anthony, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam & Eamonn Keogh (2018b). "The UEA multivariate time series classification archive, 2018". In: *arXiv preprint arXiv:1811.00075*.

Pearson, Karl (1901). "On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572.

Jolliffe, I.T. (2002). *Principal Component Analysis*. 2nd. Springer.

Golub, Gene H. & Charles F. Van Loan (2013). *Matrix Computations*. 4th. Johns Hopkins University Press.

Bingham, E. & H. Mannila (2001). "Random projection in dimensionality reduction: Applications and theoretical guarantees". In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 245–250.

Guyon, Isabelle & André Elisseeff (2003). "An introduction to variable and feature selection". In: *Journal of Machine Learning Research* 3, pp. 1157–1182.

Dziugaite, Gintare Karolina & Daniel M. Roy (2017). "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data". In: *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. arXiv: 1703.11008.

Chaudhari, Pratik, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun & Riccardo Zecchina (2017). "Entropy-SGD:

Biasing Gradient Descent Into Wide Valleys". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=B1YfAfcgl.

Keskar, Nitish Shirish, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy & Ping Tak Peter Tang (2017). "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1oyRlYgg.

Horn, Roger A. & Charles R. Johnson (1991). *Topics in Matrix Analysis*. Cambridge University Press.

Louart, Cosme & Romain Couillet (2021). "Spectral properties of sample covariance matrices arising from random matrices with independent non identically distributed columns". In: *arXiv preprint arXiv:2109.02644*.

Dobriban, Edgar & Stefan Wager (2018). "High-dimensional asymptotics of prediction: Ridge regression and classification". In: *The Annals of Statistics* 46.1, pp. 247–279.

Gerbelot, Cedric, Alia Abbara & Florent Krzakala (2022). "Asymptotic errors for teacher-student convex generalized linear models (or: How to prove Kabashima's replica formula)". In: *IEEE Transactions on Information Theory* 69.3, pp. 1824–1852.

Appendix A

SAMFormer

Roadmap. In this appendix, we provide additional background knowledge in Section A.1. The proofs of the main theoretical results are provided in Section A.2. We display the corresponding table of contents below.

A.1 Additional Background

A.1.1 Reversible Instance Normalization: RevIN

Overview. T. Kim et al. 2021 recently proposed RevIN, a reversible instance normalization to reduce the discrepancy between the distributions of training and test data. Indeed, statistical properties of real-world time series, e.g. mean and variance, can change over time, leading to non-stationary sequences. This causes a distribution shift between training and test sets for the forecasting task. The RevIN normalization scheme is now widespread in deep learning approaches for time series forecasting (S.-A. Chen et al., 2023; Yuqi Nie et al., 2023). The RevIN normalization involves trainable parameters $(\boldsymbol{\beta}, \boldsymbol{\gamma}) \in \mathbb{R}^K \times \mathbb{R}^K$ and consists of two parts: a normalization step and a symmetric denormalization step. Before presenting them, we introduce for a given input time series $\mathbf{X}^{(i)} \in \mathcal{X}$ the empirical mean $\hat{\mu}[\mathbf{X}_k^{(i)}]$ and empirical standard deviation $\hat{\sigma}^2[\mathbf{X}_k^{(i)}]$ of its k-th feature $\mathbf{X}_k^{(i)} \in \mathbb{R}^{1\times L}$ as follows:

$$\begin{cases} \hat{\mu}\left[\mathbf{X}_k^{(i)}\right] = \frac{1}{L}\sum_{t=1}^{L} \mathbf{X}_{kj}^{(i)} \\ \hat{\sigma}^2\left[\mathbf{X}_k^{(i)}\right] = \frac{1}{L}\sum_{t=1}^{L} (\mathbf{X}_{kj}^{(i)} - \hat{\mu}[\mathbf{X}_k^{(i)}])^2 \,. \end{cases} \tag{A.1}$$

The first one acts on the input sequence $\mathbf{X}^{(i)}$ and outputs the corresponding normalized sequence $\tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^{K\times L}$ such that for all k, t,

$$\tilde{\mathbf{X}}_{kt}^{(i)} = \boldsymbol{\gamma}_k \left(\frac{\mathbf{X}_{kt}^{(i)} - \hat{\mu}\left[\mathbf{X}_k^{(i)}\right]}{\sqrt{\hat{\sigma}^2\left[\mathbf{X}_k^{(i)}\right] + \varepsilon}} \right) + \boldsymbol{\beta}_k, \tag{A.2}$$

where $\varepsilon > 0$ is a small constant to avoid dividing by 0. The neural network's input is then $\tilde{\mathbf{X}}^{(i)}$, instead of $\mathbf{X}^{(i)}$. The second step is applied to the output of the neural network $\tilde{\mathbf{Y}}^{(i)}$, such that the final output considered for the forecasting is the denormalized sequence $\hat{\mathbf{Y}}^{(i)} \in \mathbb{R}^{K \times H}$ such that for all k, t,

$$\hat{\mathbf{Y}}_{kt}^{(i)} = \sqrt{\hat{\sigma}^2\left[\mathbf{X}_k^{(i)}\right] + \varepsilon} \cdot \left(\frac{\tilde{\mathbf{Y}}_{kt}^{(i)} - \boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\right) + \hat{\mu}\left[\mathbf{X}_k^{(i)}\right]. \tag{A.3}$$

As stated in T. Kim et al. 2021, $\hat{\mu}, \hat{\sigma}^2, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ contain the non-stationary information of the input sequences $\mathbf{X}^{(i)}$.

End-to-end closed form with linear model and RevIN. We consider a simple linear neural network. Formally, for any input sequence $\mathbf{X} \in \mathbb{R}^{D \times L}$, the prediction of $f_{\text{lin}} : \mathbb{R}^{D \times L} \to \mathbb{R}^{D \times H}$ simply writes

$$f_{\text{lin}}(\mathbf{X}) = \mathbf{X}\mathbf{W}. \tag{A.4}$$

When combined with RevIN, the neural network f_{lin} is not directly applied to the input sequence but after the first normalization step of RevIN (Eq. (A.2)). An interesting benefit of the simplicity of f_{lin} is that it enables us to write its prediction in closed form, even when with RevIN. The proof is deferred to Appendix A.2.4.

Proposition A.1.1 (Closed-form formulation). *For any input sequence $\mathbf{X} \in \mathbb{R}^{K \times L}$, the output of the linear model $\hat{\mathbf{Y}} = f_{\text{lin}}(\mathbf{X}) \in \mathbb{R}^{K \times H}$ has entries*

$$\hat{\mathbf{Y}}_{kt} = \hat{\mu}[\mathbf{X}_k] + \sum_{j=1}^{L} (\mathbf{X}_{kj} - \hat{\mu}[\mathbf{X}_k])\mathbf{W}_{jt} - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}\left(1 - \sum_{j=1}^{L} \mathbf{W}_{jt}\right), \tag{A.5}$$

Proposition A.1.1 highlights the fact that the k-th variable of the outputs $\hat{\mathbf{Y}}$ only depends on k-th variable of the input sequence $\mathbf{X}$. It leads to channel-independent forecasting, although we did not explicitly enforce it. (A.5) can be seen as a linear interpolation around the mean $\hat{\mu}$ with a regularization term on the network parameters $\mathbf{W}$ involving the non-stationary information $\hat{\sigma}^2, \boldsymbol{\beta}, \boldsymbol{\gamma}$. Moreover, the output sequence $\hat{\mathbf{Y}}$ can be written in a more compact and convenient matrix formulation as follows

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W} + \boldsymbol{\xi}^{(\mathrm{X},\mathrm{W},\boldsymbol{\beta},\boldsymbol{\gamma})}, \tag{A.6}$$

where $\boldsymbol{\xi}^{(\mathrm{X},\mathrm{W},\boldsymbol{\beta},\boldsymbol{\gamma})} \in \mathbb{R}^{K \times H}$ with entry $\left(\hat{\mu}[\mathbf{X}_k] - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}\right)\left(1 - \sum_{j=1}^{L} \mathbf{W}_{jt}\right)$ in the k-th row and t-th column. The proof is deferred to Appendix A.2.5. With this formulation, the predicted sequence can be seen as a sum of a linear term $\mathbf{X}\mathbf{W}$ and a residual term $\boldsymbol{\xi}^{(\mathrm{X},\mathrm{W},\boldsymbol{\beta},\boldsymbol{\gamma})}$ that takes into account the first and second moments of each variable $\mathbf{X}_k$, which is reminiscent of the linear regression model.

A.1.2 Sharpness-aware minimization (SAM)

Regularizing with the sharpness. Standard approaches consider a parametric family of models $f_{\boldsymbol{\omega}}$ and aim to find parameters $\boldsymbol{\omega}$ that minimize a training objective $\mathcal{L}_{\text{train}}(\boldsymbol{\omega})$, used as a tractable proxy to the true generalization error $\mathcal{L}_{\text{test}}(\boldsymbol{\omega})$. Most deep learning pipelines rely on first-order optimizers, e.g. SGD (Nesterov, 1983) or Adam (Kingma & Ba, 2015), that disregard higher-order information such as the curvature, despite its connection to generalization (Dziugaite & Roy, 2017; Chaudhari et al., 2017; Keskar et al., 2017). As $\mathcal{L}_{\text{train}}$ is usually non-convex in $\boldsymbol{\omega}$, with multiple local or global minima, solving $\min_{\boldsymbol{\omega}} \mathcal{L}_{\text{train}}(\boldsymbol{\omega})$ may still lead to high generalization error $\mathcal{L}_{\text{test}}(\boldsymbol{\omega})$. To alleviate this issue, Foret et al. 2021 propose to regularize the training objective with the sharpness, defined as follows

Definition A.1.2 (Sharpness, Foret et al. 2021). *For a given $\rho \geq 0$, the sharpness of $\mathcal{L}_{\text{train}}$ at $\boldsymbol{\omega}$ writes*

$$s(\boldsymbol{\omega}, \rho) \coloneqq \max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \mathcal{L}_{\text{train}}(\boldsymbol{\omega} + \boldsymbol{\epsilon}) - \mathcal{L}_{\text{train}}(\boldsymbol{\omega}). \tag{A.7}$$

Remark A.1.1 (Interpretation of ρ). *Instead of simply minimizing the training objective $\mathcal{L}_{\text{train}}$, SAM searches for parameters $\boldsymbol{\omega}$ achieving both low training loss and low curvature in a ball $\mathcal{B}(\boldsymbol{\omega}, \rho)$. The hyperparameter $\rho \geq 0$ corresponds to the size of the neighborhood on which the parameters search is done. In particular, taking $\rho = 0$ is equivalent to the usual minimization of $\mathcal{L}_{\text{train}}$.*

In particular, SAM incorporates sharpness in the learning objective, resulting in the problem of minimizing w.r.t $\boldsymbol{\omega}$

$$\mathcal{L}_{\text{train}}^{\text{SAM}}(\boldsymbol{\omega}) \coloneqq \underbrace{\max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \mathcal{L}_{\text{train}}(\boldsymbol{\omega} + \boldsymbol{\epsilon})}_{= \mathcal{L}_{\text{train}}(\boldsymbol{\omega}) + s(\boldsymbol{\omega}, \rho)} . \tag{A.8}$$

Gradient updates. As the exact solution to the inner maximization in Eq. (A.8) is hard to compute, the authors of (Foret et al., 2021) approximate it with the following first-order Taylor expansion

$$\begin{aligned} \boldsymbol{\epsilon}^*(\boldsymbol{\omega}) &\coloneqq \arg\max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \mathcal{L}_{\text{train}}(\boldsymbol{\omega} + \boldsymbol{\epsilon}) \\ &\approx \arg\max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \mathcal{L}_{\text{train}}(\boldsymbol{\omega}) + \boldsymbol{\epsilon}^\top \nabla \mathcal{L}_{\text{train}}(\boldsymbol{\omega}) \\ &= \arg\max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \boldsymbol{\epsilon}^\top \nabla \mathcal{L}_{\text{train}}(\boldsymbol{\omega}) , \end{aligned} \tag{A.9}$$

where the solution of (A.9) writes $\hat{\boldsymbol{\epsilon}}(\boldsymbol{\omega}) = \rho \frac{\nabla \mathcal{L}_{\text{train}}(\boldsymbol{\omega})}{\|\nabla \mathcal{L}_{\text{train}}(\boldsymbol{\omega})\|_2}$. It leads to the following gradient update

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \eta \nabla \mathcal{L}_{\text{train}} \left(\boldsymbol{\omega}_t + \rho \frac{\nabla \mathcal{L}_{\text{train}}(\boldsymbol{\omega})}{\|\nabla \mathcal{L}_{\text{train}}(\boldsymbol{\omega})\|_2} \right) ,$$

where η is the learning rate.

A.2 Proofs

A.2.1 Notations

To ease the readability of the proofs, we recall the following notations. We denote scalar values by regular letters (e.g., parameter λ), vectors by bold lowercase letters (e.g., vector $\mathbf{x}$), and matrices by bold capital letters (e.g., matrix $\mathbf{M}$). For a matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$, we denote by $\mathbf{M}_{i\cdot}$ its i-th row, by $\mathbf{M}_{\cdot j}$ its j-th column, by m_{ij} its entries and by $\mathbf{M}^\top$ its transpose. We denote the trace of a matrix $\mathbf{M}$ by $\mathsf{Tr}(\mathbf{M})$, its rank by $\mathsf{rank}(\mathbf{M})$ and its Frobenius norm by $\|\mathbf{M}\|_\mathsf{F}$. We denote $\boldsymbol{\sigma}(\mathbf{M}) \coloneqq (\sigma_1(\mathbf{M}), \dots, \sigma_{\tilde{n}}(\mathbf{M}))$ the vector of singular values of $\mathbf{M}$ in non-decreasing order, with $\tilde{n} = \min\{n, m\}$ and the specific notation $\sigma_{\min}(\mathbf{M}), \sigma_{\max}(\mathbf{M})$ for the minimum and maximum singular values, respectively. We denote by $\|\mathbf{M}\|_* = \sum_{i=1}^{\tilde{n}} \sigma_i(\mathbf{M})$ its nuclear norm and by $\|\mathbf{M}\|_2 = \sigma_{\max}(\mathbf{M})$ its spectral norm. When $\mathbf{M}$ is square with $n = m$, we denote $\boldsymbol{\lambda}(\mathbf{M}) \coloneqq (\lambda_1(\mathbf{M}), \dots, \lambda_n(\mathbf{M}))$ the vector of singular values of $\mathbf{M}$ in non-decreasing order and the specific notation $\lambda_{\min}(\mathbf{M}), \lambda_{\max}(\mathbf{M})$ for the minimum and maximum singular values, respectively. For a vector $\mathbf{x}$, its transpose writes $\mathbf{x}^\top$ and its usual Euclidean norm writes $\|\mathbf{x}\|$. The identity matrix of size $n \times n$ is denoted by $\mathbf{I}_n$. The vector of size n with each entry equal to 1 is denoted by $\mathbb{1}_n$. The notation $\mathbf{M} \succcurlyeq \mathbf{0}$ indicates that $\mathbf{M}$ is positive semi-definite.

A.2.2 Proof of Proposition 3.2.1

We first recall the following technical lemmas.

Lemma A.2.1. *Let $\mathbf{S} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{m\times m}$. If $\mathbf{B}$ has full rank, then*

$$\mathsf{rank}(\mathbf{SB}) = \mathsf{rank}(\mathbf{BS}) = \mathsf{rank}(\mathbf{S}).$$

Proof. Let $\mathbf{F}_1 \coloneqq \{\mathbf{Su} | \mathbf{u} \in \mathbb{R}^m\} \subset \mathbb{R}^n$ and $\mathbf{F}_2 \coloneqq \{(\mathbf{SB})\mathbf{u} | \mathbf{u} \in \mathbb{R}^m\} \subset \mathbb{R}^n$ be the vector spaces generated by the columns of $\mathbf{S}$ and $\mathbf{SB}$ respectively. By definition, the rank of a matrix is the dimension of the vector space generated by its columns (equivalently by its rows). We will show that $\mathbf{F}_1$ and $\mathbf{F}_2$ coincides. Let $\mathbf{v} \in \mathbf{F}_1$, i.e., there exists $\mathbf{u} \in \mathbb{R}^m$ such that $\mathbf{v} = \mathbf{Su}$. As $\mathbf{B}$ is full rank, the operator $\mathbf{x} \to \mathbf{Bx}$ is bijective. It follows that there always exists some $\mathbf{z} \in \mathbb{R}^m$ such that $\mathbf{u} = \mathbf{Bz}$. Then, we have

$$\mathbf{v} = \mathbf{Su} = \mathbf{S}(\mathbf{Bz}) = (\mathbf{SB})\mathbf{z},$$

which means that $\mathbf{v} \in \mathbf{F}_2$. As $\mathbf{v}$ was taken arbitrarily in $\mathbf{F}_1$, we have proved that $\mathbf{F}_1 \subset \mathbf{F}_2$. Conversely, consider $\mathbf{y} \in \mathbf{F}_2$, i.e., we can write $\mathbf{y} = (\mathbf{SB})\mathbf{z}$ for some $\mathbf{z} \in \mathbb{R}^m$. It can then be seen that

$$\mathbf{y} = (\mathbf{SB})\mathbf{z} = \mathbf{S}(\mathbf{Bz}),$$

which means that $\mathbf{y} \in \mathbf{F}_1$. Again, as $\mathbf{y}$ was taken arbitrarily, we have proved that $\mathbf{F}_1 \subset \mathbf{F}_2$. In the end, we demonstrated that $\mathbf{F}_1$ and $\mathbf{F}_2$ coincide, hence they have the same dimension. By definition of the rank, $\mathbf{S}$ and $\mathbf{SB}$ have the same rank. Similar arguments can be used to show that $\mathbf{S}$ and $\mathbf{BS}$ have the same rank, which concludes the proof. □

The next lemma is a well-known result in matrix analysis and can be found in Horn & Johnson 1991, Theorem 4.4.5. For the sake of self-consistency, we recall it below along with a sketch of the original proof.

Lemma A.2.2. *(see Horn & Johnson, 1991, Theorem 4.4.5, p. 281). Let* $\mathbf{S} \in \mathbb{R}^{n\times m}$, $\mathbf{B} = \mathbb{R}^{p\times H}$ *and* $\mathbf{C} \in \mathbb{R}^{n\times H}$. *There exists matrices* $\mathbf{Y} \in \mathbb{R}^{m\times H}$ *and* $\mathbf{Z} \in \mathbb{R}^{n\times p}$ *such that* $\mathbf{SY} - \mathbf{ZB} = \mathbf{C}$ *if, and only if,*

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{S} & \mathbf{C}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\right).$$

Proof. Assume that there exists $\mathbf{Y} \in \mathbb{R}^{m\times H}$ and $\mathbf{Z} \in \mathbb{R}^{n\times p}$ such that $\mathbf{SY} - \mathbf{ZB} = \mathbf{C}$. Recall that the following equality holds

$$\begin{bmatrix}\mathbf{S} & \mathbf{SY} - \mathbf{ZB}\\ \mathbf{0} & \mathbf{B}\end{bmatrix} = \begin{bmatrix}\mathbf{I}_m & -\mathbf{Y}\\ \mathbf{0} & \mathbf{I}_q\end{bmatrix}\begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{I}_n & \mathbf{Z}\\ \mathbf{0} & \mathbf{I}_p\end{bmatrix}. \tag{A.10}$$

Using Lemma A.2.1 on the right-hand-side of Eq. (A.10), we obtain

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{S} & \mathbf{SY} - \mathbf{ZB}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\right).$$

Using $\mathbf{SY} - \mathbf{ZB} = \mathbf{C}$ concludes the proof for the first implication of the equivalence. To prove the opposite direction, the authors of Horn & Johnson 1991 assume that

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{S} & \mathbf{C}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\right).$$

Since two matrices have the same rank if, and only if, they are equivalent, we know that there exists $\mathbf{Q} \in \mathbb{R}^{(n+p)\times(n+p)}$, $\mathbf{U} \in \mathbb{R}^{(m+q)\times(m+q)}$ non-singular such that

$$\begin{bmatrix}\mathbf{S} & \mathbf{C}\\ \mathbf{0} & \mathbf{B}\end{bmatrix} = \mathbf{Q}\begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}\end{bmatrix}\mathbf{U}. \tag{A.11}$$

The rest of the proof in Horn & Johnson 1991 is constructive and relies on Eq. (A.11) to exhibit $\mathbf{Y} \in \mathbb{R}^{m\times H}$ and $\mathbf{Z} \in \mathbb{R}^{n\times p}$ such that $\mathbf{SY} - \mathbf{ZB} = \mathbf{C}$. This concludes the proof of the equivalence. □

We now proceed to the proof of Proposition 3.2.1.

Proof. Applying Lemma A.2.2 with $\mathbf{S} = \mathbf{P}$, $\mathbf{B} = \mathbf{0}$, $\mathbf{C} = \mathbf{X}\mathbf{W}_{\text{toy}}$ and $\mathbf{W}$ in the role of $\mathbf{Y}$ ensures that there exists $\mathbf{W} \in \mathbb{R}^{L\times H}$ such that $\mathbf{P}\mathbf{W} = \mathbf{X}\mathbf{W}_{\text{toy}}$ if and only if $\text{rank}([\mathbf{P} \quad \mathbf{X}\mathbf{W}_{\text{toy}}]) = \text{rank}(\mathbf{P})$, which concludes the proof. □

A.2.3 Proof of Proposition 3.2.2

We first prove the following technical lemmas. While these lemmas are commonly used and, for most of them, straightforward to prove, they are very useful to demonstrate Proposition 3.2.2.

Lemma A.2.3 (Trace of a product of matrix). *Let $\mathbf{S}, \mathbf{B} \in \mathbb{R}^{n\times n}$ be **symmetric** matrices with $\mathbf{B}$ positive semi-definite. We have*

$$\lambda_{\min}(\mathbf{S})\,\text{Tr}(\mathbf{B}) \leq \text{Tr}(\mathbf{S}\mathbf{B}) \leq \lambda_{\max}(\mathbf{S})\,\text{Tr}(\mathbf{B}).$$

Proof. The spectral theorem ensures the existence of $\mathbf{P} \in \mathbb{R}^{n\times n}$ orthogonal, i.e., $\mathbf{P}^\top\mathbf{P} = \mathbf{P}\mathbf{P}^\top = \mathbf{I}_n$, and $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ diagonal with the eigenvalues of $\mathbf{S}$ as entries such that $\mathbf{S} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$. Benefiting from the properties of the trace operator, we have

$$\begin{aligned}
\text{Tr}(\mathbf{S}\mathbf{B}) &= \text{Tr}(\mathbf{I}_n\mathbf{S}\mathbf{B}) \\
&= \text{Tr}\left(\underbrace{\mathbf{P}\mathbf{P}^\top}_{=\mathbf{I}_n}\mathbf{S}\mathbf{B}\right) && \text{(orthogonality of } \mathbf{P}\text{)} \\
&= \text{Tr}\left(\mathbf{P}^\top\mathbf{S}\mathbf{B}\mathbf{P}\right) && \text{(cyclic property of trace)} \\
&= \text{Tr}\left(\mathbf{P}^\top\mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top\mathbf{B}\mathbf{P}\right) && \text{(Spectral theorem)} \\
&= \text{Tr}\left(\underbrace{\mathbf{P}^\top\mathbf{P}}_{=\mathbf{I}_n}\mathbf{\Lambda}\mathbf{P}^\top\mathbf{B}\mathbf{P}\right) && \text{(orthogonality of } \mathbf{P}\text{)} \\
&= \text{Tr}\left(\mathbf{\Lambda}\mathbf{P}^\top\mathbf{B}\mathbf{P}\right).
\end{aligned}$$

We introduce $\tilde{\mathbf{B}} = \mathbf{P}^\top\mathbf{B}\mathbf{P} = [\tilde{b}_{ij}]_{ij}$. It follows from the definition of $\mathbf{\Lambda}$ that

$$\text{Tr}(\mathbf{S}\mathbf{B}) = \text{Tr}\left(\mathbf{\Lambda}\mathbf{P}^\top\mathbf{B}\mathbf{P}\right) = \text{Tr}\left(\mathbf{\Lambda}\tilde{\mathbf{B}}\right) = \sum_i \lambda_i(\mathbf{S})\tilde{b}_{ii}. \tag{A.12}$$

We would like to write the $\tilde{b}_{ij}$ with respect to the p_{ij}, b_{ij} the elements of $\mathbf{P}$, $\mathbf{B}$, respectively. As $\mathbf{P}$ is orthogonal, we know that its columns $(\mathbf{e}_i)_{i=0}^n$ form an orthonormal basis of $\mathbb{R}^n$. Hence, the entry (i,j) of $\mathbf{\Lambda}\mathbf{P}^\top\mathbf{B}\mathbf{P}$, writes as follows:

$$\tilde{b}_{ij} = \sum_{kl} p_{ki}\,b_{ij}\,p_{jk}$$

$$
\begin{aligned}
&= \sum_k p_{ki} \underbrace{\left(\sum_l b_{ij} p_{jk} \right)}_{[\mathbf{Be}_j]_k} \\
&= \sum_k p_{ki} [\mathbf{Be}_j]_k \\
&= e_i^\top \mathbf{Be}_j \geq 0. && (\mathbf{B} \succcurlyeq \mathbf{0})
\end{aligned}
$$

Hence, as $\mathbf{B}$ is positive semi-definite, the $\tilde{b}_{ij}$ are nonnegative. It follows that

$$
\lambda_{\min}(\mathbf{S}) \sum_i \tilde{b}_{ii} \leq \sum_i \lambda_i(\mathbf{S}) \underbrace{\tilde{b}_{ii}}_{\geq 0} \leq \lambda_{\max}(\mathbf{S}) \sum_i \tilde{b}_{ii}. \tag{A.13}
$$

Moreover, using the definition of $\tilde{\mathbf{B}}$, the orthogonality of $\mathbf{P}$ and the cyclic property of the trace operation, we have

$$
\sum_i \tilde{b}_{ii} = \operatorname{Tr}(\tilde{\mathbf{B}}) = \operatorname{Tr}(\mathbf{P}^\top \mathbf{B} \mathbf{P}) = \operatorname{Tr}\left(\underbrace{\mathbf{P}\mathbf{P}^\top}_{=\mathbf{I}_n} \mathbf{B} \right) = \operatorname{Tr}(\mathbf{B}).
$$

Combining this last equality with Eq. (A.12) and Eq. (A.13) concludes the proof, i.e.,

$$
\lambda_{\min}(\mathbf{S}) \operatorname{Tr}(\mathbf{B}) \leq \operatorname{Tr}(\mathbf{S}\mathbf{B}) \leq \lambda_{\max}(\mathbf{S}) \operatorname{Tr}(\mathbf{B}). \tag{A.14}
$$

□

Lemma A.2.4 (Power of symmetric matrices). *Let $\mathbf{S} \in \mathbb{R}^{n \times n}$ be symmetric. The spectral theorem ensures the existence of $\mathbf{P} \in \mathbb{R}^{n \times n}$ orthogonal, i.e., $\mathbf{P}^\top \mathbf{P} = \mathbf{P}\mathbf{P}^\top = \mathbf{I}_n$, and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ diagonal with the eigenvalues of $\mathbf{S}$ as entries such that $\mathbf{S} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$. For any integer $n \geq 1$, we have*

$$
\mathbf{S}^n = \mathbf{P}\mathbf{\Lambda}^n\mathbf{P}^\top.
$$

In particular, the eigenvalues of $\mathbf{S}^n$ are equal to the eigenvalues of $\mathbf{S}$ to the power of n.

Proof. Let $n \geq 1$ be an integer. We have

$$
\begin{aligned}
\mathbf{S}^n &= \left(\mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top\right)^n \\
&= \underbrace{\mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top \times \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top \times \cdots \times \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top \times \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top}_{\times n} \\
&= \underbrace{\mathbf{P}\mathbf{\Lambda} \times \mathbf{\Lambda}\mathbf{P}^\top \ldots \mathbf{P}\mathbf{\Lambda} \times \mathbf{\Lambda}\mathbf{P}^\top}_{\times n} && \text{(orthogonality of } \mathbf{P}) \\
&= \mathbf{P} \underbrace{\mathbf{\Lambda} \times \mathbf{\Lambda} \times \cdots \times \mathbf{\Lambda} \times \mathbf{\Lambda}}_{\times n} \mathbf{P}^\top && \text{(orthogonality of } \mathbf{P}) \\
&= \mathbf{P}\mathbf{\Lambda}^n\mathbf{P}^\top.
\end{aligned}
$$

The diagonality of $\mathbf{\Lambda}$ suffices to deduct the remark on the eigenvalues of $\mathbf{S}^n$. □

Lemma A.2.5 (Case of equality between eigenvalues and singular values). *Let* $\mathbf{S} \in \mathbb{R}^{n \times n}$ *be symmetric and positive semi-definite. Then the* i*-th eigenvalue and the* i*-th singular value of* $\mathbf{S}$ *are equal, i.e., for all* $i \in [\![1, n]\!]$*, we have*

$$\lambda_i(\mathbf{S}) = \sigma_i(\mathbf{S}).$$

Proof. Let $i \in [\![1, n]\!]$. By definition of singular value, we have

$$\begin{aligned}
\sigma_i(\mathbf{S}) &\coloneqq \sqrt{\lambda_i(\mathbf{S}^\top \mathbf{S})} \\
&= \sqrt{\lambda_i(\mathbf{S}^2)} && (\mathbf{S} \text{ is symmetric}) \\
&= \sqrt{\lambda_i(\mathbf{S})^2} && (\text{Lemma A.2.4}) \\
&= |\lambda_i(\mathbf{S})| \\
&= \lambda_i(\mathbf{S}). && (\mathbf{S} \succcurlyeq \mathbf{0})
\end{aligned}$$

□

Lemma A.2.6. *Let* $\mathbf{X} \in \mathbb{R}^{D \times L}$ *be an input sequence and* $\mathbf{S} \in \mathbb{R}^{L \times L}$ *be a positive semi-definite matrix. Then,* $\mathbf{X}\mathbf{S}\mathbf{X}^\top$ *is positive semi-definite.*

Proof. It is clear that $\mathbf{X}\mathbf{S}\mathbf{X}^\top \in \mathbb{R}^{L \times L}$ is symmetric. Let $\mathbf{u} \in \mathbb{R}^L$. We have:

$$\mathbf{u}^\top \mathbf{X}\mathbf{S}\mathbf{X}^\top \mathbf{u} = \left(\mathbf{X}^\top \mathbf{u}\right)^\top \mathbf{S}\left(\mathbf{X}^\top \mathbf{u}\right) \geq 0. \qquad (\mathbf{S} \succcurlyeq \mathbf{0})$$

As $\mathbf{u}$ was arbitrarily chosen, we have proved that $\mathbf{X}\mathbf{S}\mathbf{X}^\top$ is positive semi-definite. □

We now proceed to the proof of Theorem 3.2.2.

Proof. We recall that $\mathbf{W}_Q \mathbf{W}_K^\top$ is symmetric and positive semi-definite, we have

$$\begin{aligned}
\|\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\|_* &= \operatorname{Tr}\left(\sqrt{\left(\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\right)^\top \mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top}\right) \\
&= \operatorname{Tr}\left(\sqrt{\mathbf{X}\mathbf{W}_K\mathbf{W}_Q^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top}\right) \\
&= \operatorname{Tr}\left(\sqrt{\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top}\right) && (\text{symmetry}) \\
&= \operatorname{Tr}\left(\sqrt{\left(\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\right)^2}\right) \\
&= \operatorname{Tr}\left(\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\right) && (\text{Lemma A.2.6 with } \mathbf{S} = \mathbf{W}_Q\mathbf{W}_K^\top) \\
&= \operatorname{Tr}\left(\mathbf{X}^\top\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\right). && (\text{cyclic property of the trace})
\end{aligned}$$

Using the fact that $\mathbf{X}^\top\mathbf{X}$ is positive semi-definite (Lemma A.2.6 with $\mathbf{S} = \mathbf{I}_L$), and that $\mathbf{W}_Q\mathbf{W}_K^\top$ is symmetric, Lemma A.2.3 can be applied with $\mathbf{M} = \mathbf{W}_Q\mathbf{W}_K^\top$ and $\mathbf{B} = \mathbf{X}^\top\mathbf{X}$. It leads to:

$$\|\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\|_* = \mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\big) \leq \lambda_{\max}\big(\mathbf{W}_Q\mathbf{W}_K^\top\big)\,\mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\big)\,. \qquad \text{(Lemma A.2.3)}$$

As $\mathbf{W}_Q\mathbf{W}_K^\top$ is positive semi-definite, Lemma A.2.5 ensure

$$\lambda_{\max}\big(\mathbf{W}_Q\mathbf{W}_K^\top\big) = \sigma_{\max}\big(\mathbf{W}_Q\mathbf{W}_K^\top\big) = \|\mathbf{W}_Q\mathbf{W}_K^\top\|_2$$

by definition of the spectral norm $\|\cdot\|_2$. Recalling that by definition, $\mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\big) = \|\mathbf{X}\|_{\mathrm{F}}^2$ concludes the proof, i.e.,

$$\|\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\|_* \leq \|\mathbf{W}_Q\mathbf{W}_K^\top\|_2\|\mathbf{X}\|_{\mathrm{F}}^2.$$

□

A.2.4 Proof of Proposition A.1.1

Proof. Let $k \in [\![1, K]\!]$ and $t \in [\![1, H]\!]$. We have

$$\begin{aligned}
\hat{\mathbf{Y}}_{kt} &= \sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon} \cdot \left(\frac{\tilde{\mathbf{y}}_{kt} - \boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\right) + \hat{\mu}[\mathbf{X}_k], && \text{(from (A.3))}\\
&= \sqrt{\hat{\sigma}^2[\mathbf{x}_k] + \varepsilon} \cdot \left(\frac{\sum_{j=1}^{L} \tilde{\mathbf{X}}_{kj}\mathbf{W}_{jt} - \boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\right) + \hat{\mu}[\mathbf{X}_k], && \text{(from (A.4))}\\
&= \frac{\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}}{\boldsymbol{\gamma}_k} \cdot \sum_{j=1}^{L} \tilde{\mathbf{X}}_{kj}\mathbf{W}_{jt} - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon} + \hat{\mu}[\mathbf{X}_k]\\
&= \frac{\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}}{\boldsymbol{\gamma}_k} \cdot \sum_{j=1}^{L} \left(\boldsymbol{\gamma}_k\left(\frac{\mathbf{X}_{kj} - \hat{\mu}[\mathbf{x}_k]}{\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}}\right) + \boldsymbol{\beta}_k\right)\mathbf{W}_{jt} - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{x}_k] + \varepsilon} + \hat{\mu}[\mathbf{X}_k], \\
& && \text{(from (A.2))}\\
&= \sum_{j=1}^{L}(\mathbf{X}_{kj} - \hat{\mu}[\mathbf{X}_k])\mathbf{W}_{jt} + \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}\left(\sum_{j=1}^{L}\mathbf{W}_{jt} - 1\right) + \hat{\mu}[\mathbf{X}_k]\\
&= \hat{\mu}[\mathbf{X}_k] + \sum_{j=1}^{L}(\mathbf{X}_{kj} - \hat{\mu}[\mathbf{X}_k])\mathbf{W}_{jt} - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}\left(1 - \sum_{j=1}^{L}\mathbf{W}_{jt}\right).
\end{aligned}$$

□

A.2.5 Matrix formulation of $\hat{\mathbf{Y}}$ in Eq. (A.6)

Proof. Let $k \in [\![1, K]\!]$ and $t \in [\![1, H]\!]$. From Proposition A.1.1, we have

$$\hat{\mathbf{Y}}_{kt} = \hat{\mu}[\mathbf{X}_k] + \sum_{j=1}^{L}(\mathbf{X}_{kj} - \hat{\mu}[\mathbf{X}_k])\mathbf{W}_{jt} - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k}\sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon}\left(1 - \sum_{j=1}^{L}\mathbf{W}_{jt}\right)$$

$$= \sum_{j=1}^{L} \mathbf{X}_{kj} \mathbf{W}_{jt} + \left(\hat{\mu}[\mathbf{X}_k] - \frac{\boldsymbol{\beta}_k}{\boldsymbol{\gamma}_k} \sqrt{\hat{\sigma}^2[\mathbf{X}_k] + \varepsilon} \right) \cdot \left(1 - \sum_{j=1}^{L} \mathbf{W}_{jt} \right).$$

Gathering in matrix formulation concludes the proof. □

Appendix B

On Multi-Task learning in multivariate Time Series Forecasting

Roadmap. This appendix provides the technical details omitted in the main paper. Section B.1 offers a detailed computation for $\hat{\mathbf{W}}_t$ and $\hat{\mathbf{W}}_0$. Section B.2 contains a proof of Lemma 1. Section B.3 explains the theoretical steps for deriving the training and test risks, as well as the deterministic equivalents. Section B.4 discusses the technical tools used to derive the main intuitions presented by the theory. Section B.5 focuses on the derivation of the estimations of the main quantities involved in the training and test risks. Section B.6 showcases that our theoretical framework applies very well in the multi-task regression setting. Finally, Section B.8 deals with the limitations of our approach in a non-linear setting.

Note: The proofs included in this appendix are derived from the paper presented in chapter 4 and were developed by my colleagues Malik Tiomoko and Cosme Louart.

B.1 Minimization Problem

B.1.1 Computation of $\hat{\mathbf{W}}_t$ and $\hat{\mathbf{W}}_0$

The proposed multi task regression finds $\hat{\mathbf{W}} = [\hat{\mathbf{W}}_1^\top, \ldots, \hat{\mathbf{W}}_k^\top]^\top \in \mathbb{R}^{LT \times H}$ which solves the following optimization problem using the additional assumption of relatedness between the tasks ($\mathbf{W}_t = \mathbf{W}_0 + \mathbf{V}_t$ for all tasks t):

$$\min_{(W_0, V) \in \mathbb{R}^L \times \mathbb{R}^{L \times T} \times \mathbb{R}^T} \mathcal{J}(\mathbf{W}_0, \mathbf{V}) \tag{B.1}$$

where

$$\mathcal{J}(\mathbf{W}_0, \mathbf{V}) \equiv \frac{1}{2\lambda} tr\left(\mathbf{W}_0^\top \mathbf{W}_0\right) + \frac{1}{2} \sum_{t=1}^{T} \frac{tr\left(\mathbf{V}_t^\top \mathbf{V}_t\right)}{\gamma_t} + \frac{1}{2} \sum_{t=1}^{T} tr\left(\boldsymbol{\xi}_t^\top \boldsymbol{\xi}_t\right)$$

$$\boldsymbol{\xi}_t = \mathbf{Y}^{(t)} - \frac{{\mathbf{X}^{(t)}}^\top \mathbf{W}_t}{\sqrt{TL}}, \quad \forall t \in \{1, \dots, T\}.$$

The Lagrangian introducing the lagrangian parameters for each task t, $\boldsymbol{\alpha}_t \in \mathbb{R}^{n_t \times H}$ reads as

$$\begin{aligned}\mathcal{L}(\mathbf{W}_0, \mathbf{V}_t, \boldsymbol{\xi}_t, \boldsymbol{\alpha}_t) = \frac{1}{2\lambda} tr\left(\mathbf{W}_0^\top \mathbf{W}_0\right) + \frac{1}{2}\sum_{t=1}^{T} \frac{tr\left(\mathbf{V}_t^\top \mathbf{V}_t\right)}{\gamma_t} + \frac{1}{2}\sum_{t=1} tr\left(\boldsymbol{\xi}_t^\top \boldsymbol{\xi}_t\right) \\ + \sum_{t=1}^{T} tr\left(\boldsymbol{\alpha}_t^\top \left(\mathbf{Y}^{(t)} - \frac{{\mathbf{X}^{(t)}}^\top (\mathbf{W}_0 + \mathbf{V}_t)}{\sqrt{TL}} - \boldsymbol{\xi}_t\right)\right)\end{aligned}$$

Differentiating with respect to the unknown variables $\hat{\mathbf{W}}_0$, $\hat{\mathbf{V}}_t$, $\boldsymbol{\xi}_t$, $\boldsymbol{\alpha}_t$ and $\mathbf{b}_t$, we get the following system of equation

$$\begin{aligned}\frac{1}{\lambda}\hat{\mathbf{W}}_0 - \sum_{t=1}^{T} \frac{\mathbf{X}^{(t)}\boldsymbol{\alpha}_t}{\sqrt{TL}} &= 0 \\ \frac{1}{\gamma_t}\hat{\mathbf{V}}_t - \frac{\mathbf{X}^{(t)}\boldsymbol{\alpha}_t}{\sqrt{TL}} &= 0 \\ \boldsymbol{\xi}_t - \boldsymbol{\alpha}_t &= 0 \\ \mathbf{Y}^{(t)} - \frac{{\mathbf{X}^{(t)}}^\top \hat{\mathbf{W}}_0}{\sqrt{TL}} - \frac{{\mathbf{X}^{(t)}}^\top \hat{\mathbf{V}}_t}{\sqrt{TL}} - \boldsymbol{\xi}_t &= 0\end{aligned}$$

Plugging the expression of $\hat{\mathbf{W}}_0$, $\hat{\mathbf{V}}_t$ and $\boldsymbol{\xi}_t$ into the expression of $\mathbf{Y}^{(t)}$ gives

$$\mathbf{Y}^{(t)} = \lambda \sum_{t=1}^{T} \frac{{\mathbf{X}^{(t)}}^\top \mathbf{X}^{(t)}}{TL}\boldsymbol{\alpha}_t + \gamma_t \frac{{\mathbf{X}^{(t)}}^\top \mathbf{X}^{(t)}}{TL}\boldsymbol{\alpha}_t + \boldsymbol{\alpha}_t$$

which can be rewritten as

$$\mathbf{Y}^{(t)} = (\lambda + \gamma_t)\frac{{\mathbf{X}^{(t)}}^\top \mathbf{X}^{(t)}}{TL}\boldsymbol{\alpha}_t + \lambda \sum_{v \neq t} \frac{{\mathbf{X}^{(t)}}^\top \mathbf{X}^{(v)}}{TL}\boldsymbol{\alpha}_v + \boldsymbol{\alpha}_t$$

With $\mathbf{Y} = [{\mathbf{Y}^{(1)}}^\top, \dots, {\mathbf{Y}^{(T)}}^\top]^\top \in \mathbb{R}^{n \times H}$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^\top, \dots, \boldsymbol{\alpha}_k^\top]^\top \in \mathbb{R}^{n \times H}$, and $\mathbf{Z} = \sum_{t=1}^{T} \mathbf{e}_t^{[T]} {\mathbf{e}_t^{[T]}}^\top \otimes \mathbf{X}^{(t)} \in \mathbb{R}^{TL \times n}$, this system of equations can be written under the following compact matrix form:

$$\mathbf{Q}^{-1}\boldsymbol{\alpha} = \mathbf{Y}$$

with $\mathbf{Q} = \left(\frac{\mathsf{Z}^\top \mathsf{A} \mathsf{Z}}{TL} + \mathbf{I}_n\right)^{-1} \in \mathbb{R}^{n \times n}$, and $\mathbf{A} = \left(\mathcal{D}_{\boldsymbol{\gamma}} + \lambda \mathbb{1}_T \mathbb{1}_T^\top\right) \otimes \mathbf{I}_d \in \mathbb{R}^{TL \times TL}$.

Solving for $\boldsymbol{\alpha}$ then gives:

$$\boldsymbol{\alpha} = \mathbf{Q}\mathbf{Y}$$

Moreover, using $\hat{\mathbf{W}}_t = \hat{\mathbf{W}}_0 + \hat{\mathbf{V}}_t$, the expression of $\mathbf{W}_t$ becomes:

$$\hat{\mathbf{W}}_t = \left(\mathbf{e}_t^{[T]^\top} \otimes \mathbf{I}_L\right) \frac{\mathbf{AZ}\boldsymbol{\alpha}}{\sqrt{TL}},$$
$$\hat{\mathbf{W}}_0 = \left(\mathbb{1}_T^\top \otimes \lambda \mathbf{I}_L\right) \frac{\mathbf{Z}\boldsymbol{\alpha}}{\sqrt{TL}}.$$

B.2 Lemma 1 and proof with Random Matrix Theory

B.2.1 Lemma 1

Lemma 1 (Deterministic equivalents for $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{Q}}\mathbf{M}\tilde{\mathbf{Q}}$ and $\mathbf{Q}^2$ for any $\mathbf{M} \in \mathbb{R}^{n\times n}$). *Under the concentrated random vector assumption for each feature vector $\mathbf{x}_i^{(t)}$ and under the growth rate assumption (Assumption 4.2.2), for any deterministic $\mathbf{M} \in \mathbb{R}^{n\times n}$, we have the following convergence:*

$$\tilde{\mathbf{Q}} \leftrightarrow \bar{\tilde{\mathbf{Q}}}, \qquad \tilde{\mathbf{Q}}\mathbf{M}\tilde{\mathbf{Q}} \leftrightarrow \bar{\tilde{\mathbf{Q}}}_2(\mathbf{M}), \qquad \mathbf{Q}^2 \leftrightarrow \bar{\mathbf{Q}}_2$$

where $\bar{\tilde{\mathbf{Q}}}_2$, $\bar{\tilde{\mathbf{Q}}}$ and $\bar{\mathbf{Q}}_2$ are defined as follows

$$\bar{\tilde{\mathbf{Q}}} = \left(\sum_{t=1}^{T} \frac{c_0 \mathbf{C}^{(t)}}{1+\delta_t} + \mathbf{I}_{TL}\right)^{-1}, \quad \delta_t = \frac{1}{TL} tr\left(\boldsymbol{\Sigma}^{(t)} \bar{\tilde{\mathbf{Q}}}\right), \quad \mathbf{C}^{(t)} = \mathbf{A}^{\frac{1}{2}}\left(\mathbf{e}_t^{[T]} \otimes \boldsymbol{\Sigma}^{(t)}\right)\mathbf{A}^{\frac{1}{2}}$$

$$\bar{\tilde{\mathbf{Q}}}_2(\mathbf{M}) = \bar{\tilde{\mathbf{Q}}}\mathbf{M}\bar{\tilde{\mathbf{Q}}} + \frac{1}{TL}\sum_{t=1}^{T} \frac{d_t}{1+\delta_t} \bar{\tilde{\mathbf{Q}}}\mathbf{C}^{(t)}\bar{\tilde{\mathbf{Q}}}, \qquad \mathbf{d} = \left(\mathbf{I}_T - \frac{1}{TL}\Psi\right)^{-1}\Psi(\mathbf{M}) \in \mathbb{R}^T$$

$$\bar{\mathbf{Q}}_2 = \mathbf{I}_n - Diag_{t\in[T]}(v_t \mathbf{I}_{n_t}), \quad v_t = \frac{1}{TL}\frac{tr(\mathbf{C}^{(t)}\bar{\tilde{\mathbf{Q}}})}{(1+\delta_t)^2} + \frac{1}{TL}\frac{tr\left(\mathbf{C}^{(t)}\bar{\tilde{\mathbf{Q}}}_2(\mathbf{I}_n)\right)}{(1+\delta_t)^2}$$

where

$$\Psi(M) = \left(\frac{n_t}{TL}\frac{tr\left(\mathbf{C}^{(t)}\bar{\tilde{\mathbf{Q}}}\mathbf{M}\bar{\tilde{\mathbf{Q}}}\right)}{1+\delta_t}\right)_{t\in[T]} \in \mathbb{R}^T, \qquad \Psi = \left(\frac{n_t}{TL}\frac{tr\left(\mathbf{C}^{(t)}\bar{\tilde{\mathbf{Q}}}\mathbf{C}^{(t')}\bar{\tilde{\mathbf{Q}}}\right)}{(1+\delta_t)(1+\delta_{t'})}\right)_{t,t'\in[T]} \in \mathbb{R}^{T\times T}$$

B.2.2 Deterministic equivalent of the resolvent $\tilde{\mathbf{Q}}$

The evaluation of the expectation of linear forms on $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{Q}}^2$ can be found in the literature. To find a result that meets exactly our setting, we will citep (Louart & Couillet, 2021) that is a bit more general since it treats cases where $\mathbb{E}[x_i^{(t)}] \neq 0$ for $t \in [T]$ and $i \in [n_t]$. Unlike the main paper, and to be more general, the study presented below is "quasi asymptotic" meaning that the results are true for finite value of d, n. Let us first rewrite the general

required hypotheses, adapting them to our setting. For that purpose, we consider in the rest of this paper a certain asymptotic $I \subset \{(d,n), d \in \mathbb{N}, n \in \mathbb{N}\} = \mathbb{N}^2$ satisfying:

$$\{L, \exists\, n \in \mathbb{N} :\ (d,n) \in I\} = \mathbb{N} \qquad \text{and} \qquad \{n, \exists\, d \in \mathbb{N} :\ (d,n) \in I\} = \mathbb{N}.$$

such that n and d can tend to ∞ but with some constraint that is given in the first item of Assumption 1 below. Given two sequences $(a_{L,n})_{L,n\in I}, (b_{L,n})_{L,n\in I} > 0$, the notation $a_{L,n} \leq O(b_{L,n})$ (or $a \leq O(b)$) means that there exists a constant $C > 0$ such that for all $(d,n) \in I$, $a_{L,n} \leq C b_{L,n}$.

Assumption 1. *There exists some constants $C, c > 0$ independent such that:*

- $n \leq O(d)$
- $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n) \in \mathbb{R}^{TL\times n}$ *has independent columns*
- *for any $(d,n) \in I$, and any $f : \mathbb{R}^{TL\times n} \to \mathbb{R}$ 1-Lipschitz for the euclidean norm:*

$$\mathbb{P}\left(|f(\mathbf{Z}) - \mathbb{E}[f(\mathbf{Z})]| \geq t\right) \leq Ce^{-ct^2}.$$

- $\forall i \in \{n, \exists d \in \mathbb{N}, (d,n) \in I\}$: $\|\mathbb{E}[\mathbf{z}_i]\| \leq O(1)$.

Theorem 1 ((Louart & Couillet, 2021), Theorem 0.9.). *Given $T \in \mathbb{N}$, $\mathbf{Z} \in \mathbb{R}^{TL\times n}$ and two deterministic $A \in \mathbb{R}^{TL\times TL}$, we note $\tilde{\mathbf{Q}} \equiv (\frac{1}{TL}\mathbf{A}^{\frac{1}{2}}\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^{\frac{1}{2}} + I_{TL})^{-1}$. If $\mathbf{Z}$ satisfies Assumption 1 and $\mathbf{M} \in \mathbb{R}^{TL\times TL}$ is a deterministic matrix satisfying $\|\mathbf{M}\|_F \leq 1$, one has the concentration:*

$$\mathbb{P}\left(\left|tr(M\tilde{\mathbf{Q}}) - tr(M\bar{\bar{\mathbf{Q}}}_{\boldsymbol{\delta}(\mathbf{S})}(\mathbf{S}))\right| \geq t\right) \leq Ce^{-ct^2},$$

where $\mathbf{S} = (\mathbf{S}_1, \dots, \mathbf{S}_n) = (\mathbb{E}[\mathbf{z}_1\mathbf{z}_1^\top], \dots, \mathbb{E}[\mathbf{z}_n\mathbf{z}_n^\top])$, for $\boldsymbol{\delta} \in \mathbb{R}^n$, $\bar{\bar{\mathbf{Q}}}_{\boldsymbol{\delta}}$ is defined as:

$$\bar{\bar{\mathbf{Q}}}_{\boldsymbol{\delta}}(\mathbf{S}) = \left(\frac{1}{TL}\sum_{i\in[n]} \frac{\mathbf{A}^{\frac{1}{2}}\mathbf{S}_i\mathbf{A}^{\frac{1}{2}}}{1+\boldsymbol{\delta}_i} + I_{TL}\right)^{-1},$$

and $\boldsymbol{\delta}(\mathbf{S})$ is the unique solution to the system of equations:

$$\forall i \in [n] : \quad \boldsymbol{\delta}(\mathbf{S})_i = \frac{1}{n} tr\left(\mathbf{A}^{\frac{1}{2}}\mathbf{S}_i\mathbf{A}^{\frac{1}{2}}\bar{\bar{\mathbf{Q}}}_{\boldsymbol{\delta}(\mathbf{S})}\right).$$

We end this subsection with some results that will be useful for next subsection on the estimation of bilinear forms on $\bar{\bar{\mathbf{Q}}}$.

Lemma 2 ((Louart & Couillet, 2021), Lemmas 4.2, 4.6). *Under the setting of Theorem 1, given a deterministic vector $\mathbf{u} \in \mathbb{R}^{TL}$ such that $\|\mathbf{u}\| \leq O(1)$ and two deterministic matrices $\mathbf{U}, \mathbf{V}$ such that $\|\mathbf{U}\|, \|\mathbf{V}\| \leq O(1)$ and a power $r > 0$, $r \leq O(1)$:*

- $\mathbb{E}\left[\left|\mathbf{u}^\top\mathbf{U}\tilde{\mathbf{Q}}_{-i}\mathbf{V}\mathbf{z}_i\right|^r\right] \leq O(1)$
- $\mathbb{E}\left[\left|\frac{1}{TL}\mathbf{z}_i^\top\mathbf{U}\tilde{\mathbf{Q}}_{-i}\mathbf{V}\mathbf{z}_i - \mathbb{E}\left[\frac{1}{TL}tr\left(\Sigma_i\mathbf{U}\bar{\bar{\mathbf{Q}}}\mathbf{B}\right)\right]\right|^r\right] \leq O\left(\frac{1}{L^{\frac{r}{2}}}\right)$.

B.2.3 Deterministic equivalent of bilinear forms of the resolvent

To simplify the expression of the following theorem, we take $\mathbf{A} = \mathbf{I}_{TL}$. One can replace $\mathbf{Z}$ with $\mathbf{A}^{\frac{1}{2}}\mathbf{Z}$ to retrieve the result necessary for the main paper.

Theorem 2. *Under the setting of Theorem 1, with $\mathbf{A} = \mathbf{I}_{TL}$, one can estimate for any deterministic matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{TL}$ such that $\|\mathbf{U}\|, \|\mathbf{V}\| \leq O(1)$ and any deterministic vector $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{TL}$ such that $\|\mathbf{u}\|, \|\mathbf{v}\| \leq 1$, if one notes $\mathbf{B} = \frac{1}{TL}\mathbf{V}$ or $\mathbf{B} = \mathbf{u}\mathbf{v}^\top$, one can estimate:*

$$\left| \mathbb{E}\left[tr(\mathbf{B}\tilde{\mathbf{Q}}\mathbf{U}\tilde{\mathbf{Q}}) \right] - \Psi(\mathbf{U}, \mathbf{B}) - \frac{1}{TL}\Psi(\mathbf{U})^\top \left(\mathbf{I}_n - \frac{1}{TL}\Psi \right)^{-1} \Psi(\mathbf{B}) \right| \leq O\left(\frac{1}{\sqrt{L}} \right) \tag{B.2}$$

where we noted:

- $\bar{\tilde{\mathbf{Q}}} \equiv \bar{\tilde{\mathbf{Q}}}_{\boldsymbol{\delta}}(\mathbf{S}), \boldsymbol{\delta} = \boldsymbol{\delta}(\mathbf{S})$,
- $\Psi \equiv \frac{1}{TL}\left(\frac{tr\left(\mathbf{S}_i \bar{\tilde{\mathbf{Q}}} \mathbf{S}_j \bar{\tilde{\mathbf{Q}}}\right)}{(1+\delta_i)(1+\delta_j)} \right)_{i,j\in[n]} \in \mathbb{R}^{n,n}$
- $\forall \mathbf{U} \in \mathbb{R}^{n\times n} : \Psi(\mathbf{U}) \equiv \frac{1}{TL}\left(\frac{tr\left(\mathbf{U} \bar{\tilde{\mathbf{Q}}} \mathbf{S}_i \bar{\tilde{\mathbf{Q}}}\right)}{1+\delta_i} \right)_{i\in[n]} \in \mathbb{R}^n$
- $\forall \mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times n} : \Psi(\mathbf{U}, \mathbf{V}) \equiv \frac{1}{TL} tr\left(\mathbf{U}\bar{\tilde{\mathbf{Q}}}\mathbf{V}\bar{\tilde{\mathbf{Q}}} \right) \in \mathbb{R}$

If there exist $T < n$ dinstinct matrices $\mathbf{C}_1, \ldots, \mathbf{C}_T$ such that:

$$\{\mathbf{S}_1, \ldots, \mathbf{S}_n\} = \{\mathbf{C}_1, \ldots, \mathbf{C}_T\},$$

and if we denote $\forall t \in [T]$ $n_t = \#\{i \in [n] \mid \mathbf{S}_i = \mathbf{C}_t\}$ and:

$$P \equiv \left(I_T - \left(\frac{n_t n_v}{(TL)^2} \frac{tr\left(\mathbf{S}_t \bar{\tilde{\mathbf{Q}}} \mathbf{S}_v \bar{\tilde{\mathbf{Q}}}\right)}{(1+\delta_t)(1+\delta_v)} \right)_{t,v\in[T]} \right)^{-1} \in \mathbb{R}^{T,T}$$

$$\forall \mathbf{U} \in \mathbb{R}^{TL\times TL} : \quad \bar{\tilde{\mathbf{Q}}}_2(\mathbf{U}) \equiv \bar{\tilde{\mathbf{Q}}}\mathbf{U}\bar{\tilde{\mathbf{Q}}} + \frac{1}{(TL)^2} \sum_{t,v=1}^{T} \frac{tr(\mathbf{S}_t \bar{\tilde{\mathbf{Q}}}\mathbf{U}\bar{\tilde{\mathbf{Q}}}) P_{t,v} \bar{\tilde{\mathbf{Q}}}\mathbf{S}_v\bar{\tilde{\mathbf{Q}}}}{(1+\delta_t)(1+\delta_v)},$$

the result of Theorem 2 rewrites:

$$\left\| \mathbb{E}\left[\tilde{\mathbf{Q}}\mathbf{U}\tilde{\mathbf{Q}} \right] - \bar{\tilde{\mathbf{Q}}}_2(\mathbf{U}) \right\| \leq O\left(\frac{1}{\sqrt{L}} \right) \tag{B.3}$$

Proof. Given $i \in [n]$, let us note $\mathbf{Z}_{-i} = (\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, 0, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n)$ and $\tilde{\mathbf{Q}}_{-i} = (\frac{1}{TL}\mathbf{Z}_{-i}\mathbf{Z}_{-i}^\top + \mathbf{I}_{TL})^{-1}$, then we have the identity:

$$\tilde{\mathbf{Q}} - \tilde{\mathbf{Q}}_{-i} = \frac{1}{TL}\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i} \qquad \text{and} \qquad \tilde{\mathbf{Q}}\mathbf{z}_i = \frac{\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i}{1 + \frac{1}{TL}\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i}. \tag{B.4}$$

Given $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{TL}$, such that $\|\mathbf{u}\|, \|\mathbf{v}\| \leq 1$, let us express:

$$\mathbb{E}\left[\frac{1}{TL}\mathbf{u}^\top\left(\tilde{\mathbf{Q}}-\bar{\tilde{\mathbf{Q}}}\right)\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\left(\frac{\mathbf{S}_i}{1+\boldsymbol{\delta}_i}-\mathbf{z}_i\mathbf{z}_i^\top\right)\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] \tag{B.5}$$

$$\tag{B.6}$$

First, given $i \in [n]$, let us estimate thanks to (B.4):

$$\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] = \mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] - \frac{1}{TL}\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right]$$

Hölder inequality combined with Lemma 2 allows us to bound:

$$\frac{1}{TL}\left|\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right]\right| \leq \frac{1}{TL}\mathbb{E}\left[\left|\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{z}_i\right|^2\right]^{\frac{1}{2}}\mathbb{E}\left[\left|\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right|^2\right]^{\frac{1}{2}} \leq O\left(\frac{1}{L}\right),$$

one can thus deduce:

$$\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] = \mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] + O\left(\frac{1}{L}\right) = \mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}_{-i}\mathbf{v}\right] + O\left(\frac{1}{L}\right). \tag{B.7}$$

Second, one can also estimate thanks to Lemma B.4:

$$\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] = \mathbb{E}\left[\frac{\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}}{1+\frac{1}{TL}\mathbf{z}_i^\top\mathbf{Q}_{-i}\mathbf{z}_i}\right] = \mathbb{E}\left[\frac{\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}}{1+\delta_i}\right] + O\left(\frac{1}{\sqrt{d}}\right),$$

again thanks to Hölder inequality combined with Lemma 2 that allow us to bound:

$$\begin{aligned}\mathbb{E}&\left[\left|\frac{\delta_i-\frac{1}{TL}\mathbf{z}_i^\top\mathbf{Q}_{-i}\mathbf{z}_i}{(1+\delta_i)\left(1+\frac{1}{TL}\mathbf{z}_i^\top\mathbf{Q}_{-i}\mathbf{z}_i\right)}\right||\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}|\right]\\ &\leq \mathbb{E}\left[\left|\delta_i-\frac{1}{TL}\mathbf{z}_i^\top\mathbf{Q}_{-i}\mathbf{z}_i\right|^2\right]^{\frac{1}{2}}\mathbb{E}\left[|\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}|^2\right]^{\frac{1}{2}} \leq O\left(\frac{1}{\sqrt{d}}\right),\end{aligned}$$

The independence between $\mathbf{z}_i$ and $\tilde{\mathbf{Q}}_{-i}$ (and $\bar{\tilde{\mathbf{Q}}}$) then allow us to deduce (again with formula (B.4)):

$$\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] = \mathbb{E}\left[\frac{\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}_{-i}\mathbf{v}}{1+\delta_i}\right] + \frac{1}{TL}\mathbb{E}\left[\frac{\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{v}}{1+\delta_i}\right] + O\left(\frac{1}{\sqrt{d}}\right). \tag{B.8}$$

Let us inject (B.7) and (B.8) in (B.5) to obtain (again with an application of Hölder inequality and Lemma 2 that we do not detail this time):

$$\mathbb{E}\left[\mathbf{u}^\top\tilde{\mathbf{Q}}\left(\frac{\mathbf{S}_i}{1+\boldsymbol{\delta}_i}-\mathbf{z}_i\mathbf{z}_i^\top\right)\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{v}\right] = \frac{1}{TL}\mathbb{E}\left[\frac{\mathbf{u}^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\mathbf{z}_i^\top\bar{\tilde{\mathbf{Q}}}\mathbf{U}\tilde{\mathbf{Q}}\mathbf{z}_i\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{v}}{\left(1+\frac{1}{n}\mathbf{z}_i^\top\tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\right)^2}\right] + O\left(\frac{1}{\sqrt{L}}\right),$$

$$= \frac{1}{TL}\frac{\mathbb{E}\left[\mathbf{u}^\top \tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\tilde{\mathbf{Q}}_{-i}\mathbf{v}\right]}{(1+\delta_i)^2} tr\left(\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\bar{\tilde{\mathbf{Q}}}\right) + O\left(\frac{1}{\sqrt{L}}\right),$$

Putting all the estimations together, one finally obtains:

$$\left\| \mathbb{E}\left[\tilde{\mathbf{Q}}\mathbf{U}\tilde{\mathbf{Q}}\right] - \mathbb{E}\left[\bar{\tilde{\mathbf{Q}}}\mathbf{U}\bar{\tilde{\mathbf{Q}}}\right] - \frac{1}{(TL)^2}\sum_{i=1}^{n}\frac{tr\left(\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\mathbf{U}\bar{\tilde{\mathbf{Q}}}\right)}{(1+\delta_i)^2}\mathbb{E}\left[\tilde{\mathbf{Q}}_{-i}\mathbf{S}_i\tilde{\mathbf{Q}}_{-i}\right] \right\| \leq O\left(\frac{1}{\sqrt{L}}\right) \quad \text{(B.9)}$$

One then see that if we introduce for any $\mathbf{V} \in \mathbb{R}^{n\times n}$ the block matrices:

- $\theta = \frac{1}{TL}\left(\frac{\mathbb{E}\left[tr(\mathsf{S}_j\tilde{\mathsf{Q}}\mathsf{S}_i\tilde{\mathsf{Q}}^\gamma)\right]}{(1+\delta_i)(1+\delta_j)}\right)_{i,j\in[n]} \in \mathbb{R}^{n\times n}$
- $\theta(\mathbf{V}) = \frac{1}{TL}\left(\frac{\mathbb{E}\left[tr(\mathsf{V}\tilde{\mathsf{Q}}\mathsf{S}_i\tilde{\mathsf{Q}}^\gamma)\right]}{1+\delta_i}\right)_{i\in[n]} \in \mathbb{R}^n$,
- $\theta(\mathbf{U},\mathbf{V}) = \frac{1}{TL}\mathbb{E}\left[tr(\mathbf{V}\tilde{\mathbf{Q}}\mathbf{U}\tilde{\mathbf{Q}}^\gamma)\right] \in \mathbb{R}$,

then, if $\|\mathbf{V}\| \leq O(1)$, multiplying (B.9) with $\mathbf{V}$ and taking the trace leads to:

$$\theta(\mathbf{U},\mathbf{V}) = \Psi(\mathbf{U},\mathbf{V}) + \frac{1}{TL}\Psi(\mathbf{U})^\top\theta(\mathbf{V}) + O\left(\frac{1}{\sqrt{L}}\right), \quad \text{(B.10)}$$

Now, taking $\mathbf{U} = \frac{\mathsf{S}_1}{1+\delta_1}, \ldots, \frac{\mathsf{S}_n}{1+\delta_n}$, one gets the vectorial equation:

$$\theta(\mathbf{V}) = \Psi(\mathbf{V}) + \frac{1}{TL}\Psi\theta(\mathbf{V}) + O\left(\frac{1}{\sqrt{L}}\right),$$

When $(I_{TL} - \frac{1}{TL}\Psi)$ is invertible, one gets $\theta(\mathbf{V}) = (I_{TL} - \frac{1}{TL}\Psi)^{-1}\Psi(\mathbf{V}) + O\left(\frac{1}{\sqrt{L}}\right)$, and combining with (B.10), one finally obtains:

$$\theta(\mathbf{U},\mathbf{V}) = \Psi(\mathbf{U},\mathbf{V}) + \frac{1}{TL}\Psi(\mathbf{U})^\top(I_{TL} - \frac{1}{TL}\Psi)^{-1}\Psi(\mathbf{V}) + O\left(\frac{1}{\sqrt{L}}\right).$$

□

B.2.4 Estimation of the deterministic equivalent of $\mathbf{Q}^2$

Theorem 3. *Under the setting of Theorem 2, one can estimate:*

$$\left\|\mathbb{E}\left[\mathbf{Q}^2\right] - \mathbf{I}_n + \mathcal{D}_v\right\| \leq O\left(\frac{1}{\sqrt{L}}\right), \quad \text{(B.11)}$$

with, $\forall i \in [n]$*:*

$$v_i \equiv \frac{1}{TL}\frac{tr\left(\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\right)}{(1+\delta_i)^2} + \frac{1}{TL}\frac{tr\left(\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}_2(\mathbf{I}_n)\right)}{(1+\delta_i)^2}$$

Proof. The justifications are generally the same as in the proof of Theorem 2, we will thus allow ourselves to be quicker in this proof.

Using the definition of $\mathbf{Q} = \left(\frac{Z^\top AZ}{TL} + \mathbf{I}_n\right)^{-1}$, we have that

$$\frac{\mathbf{Z}^\top \mathbf{Z}}{TL}\mathbf{Q} = \left(\frac{\mathbf{Z}^\top \mathbf{Z}}{TL} + \mathbf{I}_n - \mathbf{I}_n\right)\left(\frac{\mathbf{Z}^\top \mathbf{Z}}{TL} + \mathbf{I}_n\right)^{-1} = \mathbf{I}_n - \mathbf{Q} \tag{B.12}$$

and one can then let appear $\tilde{\mathbf{Q}}$ thanks to the relation:

$$\mathbf{ZQ} = \tilde{\mathbf{Q}}\mathbf{Z}, \tag{B.13}$$

that finally gives us:

$$\mathbf{Q} = \mathbf{I}_n - \frac{1}{TL}\mathbf{Z}^\top \mathbf{ZQ} = \mathbf{I}_n - \frac{1}{TL}\mathbf{Z}^\top \tilde{\mathbf{Q}}\mathbf{Z}$$

One can then express:

$$\begin{aligned}\mathbf{Q}^2 &= \mathbf{I}_n - \frac{2}{TL}\mathbf{Z}^\top \tilde{\mathbf{Q}}\mathbf{Z} + \frac{1}{(TL)^2}\mathbf{Z}^\top \tilde{\mathbf{Q}}\mathbf{Z}\mathbf{Z}^\top \tilde{\mathbf{Q}}\mathbf{Z} \\ &= \mathbf{I}_n - \frac{1}{TL}\mathbf{Z}^\top \tilde{\mathbf{Q}}\mathbf{Z} - \frac{1}{TL}\mathbf{Z}^\top \tilde{\mathbf{Q}}^2\mathbf{Z}.\end{aligned}$$

Given $i, j \in [n]$, $i \neq j$, let us first estimate (thanks to Hölder inequality and Lemma 2):

$$\frac{1}{TL}\mathbb{E}\left[\mathbf{z}_i^\top \tilde{\mathbf{Q}}\mathbf{z}_j\right] = \frac{1}{TL}\frac{\mathbb{E}\left[\mathbf{z}_i^\top \tilde{\mathbf{Q}}_{-i,j}\mathbf{z}_j\right]}{(1+\delta_i)(1+\delta_j)} + O\left(\frac{1}{\sqrt{d}}\right) \leq O\left(\frac{1}{\sqrt{d}}\right),$$

since $\mathbb{E}[z_i] = \mathbb{E}[z_j] = 0$. Now, we consider the case $j = i$ to get:

$$\frac{1}{TL}\mathbb{E}\left[\mathbf{z}_i^\top \tilde{\mathbf{Q}}\mathbf{z}_i\right] = \frac{1}{TL}\frac{\mathbb{E}\left[\mathbf{z}_i^\top \tilde{\mathbf{Q}}_{-i}\mathbf{z}_i\right]}{(1+\delta_i)^2} + O\left(\frac{1}{\sqrt{d}}\right) = \frac{1}{TL}\frac{tr\left(\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}\right)}{(1+\delta_i)^2} + O\left(\frac{1}{\sqrt{d}}\right).$$

As before, we know that $\frac{1}{TL}\mathbb{E}\left[\mathbf{z}_i^\top \tilde{\mathbf{Q}}\mathbf{z}_j\right] \leq O\left(\frac{1}{\sqrt{d}}\right)$ if $i \neq j$. Considering $i \in [n]$, we thus are left to estimate:

$$\frac{1}{TL}\mathbb{E}\left[\mathbf{z}_i^\top \tilde{\mathbf{Q}}^2\mathbf{z}_j\right] = \frac{1}{TL}\frac{tr\left(\mathbf{S}_i\bar{\tilde{\mathbf{Q}}}_2(\mathbf{I}_n)\right)}{(1+\delta_i)^2} + O\left(\frac{1}{\sqrt{d}}\right)$$

□

B.3 Risk Estimation (Proof of Theorem 4.3.1)

B.3.1 Test Risk

The expected value of the MSE of the test data $\mathbf{x} \in \mathbb{R}^{T\times TL}$ concatenating the feature vector of all the tasks with the corresponding response variable $\mathbf{y} \in \mathbb{R}^{T\times TH}$ reads as

$$
\begin{aligned}
\mathcal{R}^{\infty}_{test} &= \frac{1}{T}\mathbb{E}[\|\mathbf{y} - g(\mathbf{x})\|_2^2] \\
&= \frac{1}{T}\mathbb{E}\left[\|\frac{\mathbf{x}^\top\mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon} - \frac{\mathbf{x}^\top\mathbf{AZQY}}{TL}\|_2^2\right] \\
&= \frac{1}{T}\mathbb{E}\left[\|\frac{\mathbf{x}^\top\mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon} - \frac{\mathbf{x}^\top\mathbf{AZQ}(\frac{\mathbf{Z}^\top\mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon})}{TL}\|_2^2\right] \\
&= \frac{1}{T}\mathbb{E}\left[\|\frac{\mathbf{x}^\top\mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon} - \frac{\mathbf{x}^\top\mathbf{AZQZ}^\top\mathbf{W}}{TL\sqrt{TL}} - \frac{\mathbf{x}^\top\mathbf{AZQ}\boldsymbol{\varepsilon}}{TL}\|_2^2\right] \\
&= \frac{1}{T}\mathbb{E}\left[\frac{tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{W}\right)}{TL} - \frac{2tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{AZQZ}^\top\mathbf{W}\right)}{(TL)^2} + tr\left(\boldsymbol{\varepsilon}^\top\boldsymbol{\varepsilon}\right) + \frac{tr\left(\mathbf{W}^\top\mathbf{ZQZ}^\top\mathbf{A}\boldsymbol{\Sigma}\mathbf{AZQZ}^\top\mathbf{W}\right)}{(TL)^3} + \right. \\
&\quad \left. \frac{tr\left(\boldsymbol{\varepsilon}^\top\mathbf{QZ}^\top\mathbf{A}\boldsymbol{\Sigma}\mathbf{AZQ}\boldsymbol{\varepsilon}\right)}{(TL)^2}\right] \\
&= \frac{1}{T}\mathbb{E}\left[\frac{tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{W}\right)}{TL} - \frac{2tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{A}^{\frac{1}{2}}(\mathbf{I}_{TL} - \tilde{\mathbf{Q}})\mathbf{A}^{-\frac{1}{2}}\mathbf{W}\right)}{TL} + \right. \\
&\quad \left. tr\left(\boldsymbol{\varepsilon}^\top\boldsymbol{\varepsilon}\right) + \frac{tr\left(\mathbf{W}^\top\mathbf{ZQZ}^\top\mathbf{A}\boldsymbol{\Sigma}\mathbf{AZQZ}^\top\mathbf{W}\right)}{(TL)^3} + \frac{tr\left(\boldsymbol{\varepsilon}^\top\mathbf{QZ}^\top\mathbf{A}\boldsymbol{\Sigma}\mathbf{AZQ}\boldsymbol{\varepsilon}\right)}{(TL)^2}\right] \\
&= \frac{tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{W}\right)}{TL} - 2\frac{tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{A}^{\frac{1}{2}}(\mathbf{I}_{TL} - \tilde{\mathbf{Q}})\mathbf{A}^{-\frac{1}{2}}\mathbf{W}\right)}{TL} + tr\left(\boldsymbol{\varepsilon}^\top\boldsymbol{\varepsilon}\right) + \frac{tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}\mathbf{W}\right)}{TL} \\
&\quad - 2\frac{tr\left(\mathbf{W}^\top\boldsymbol{\Sigma}A^{\frac{1}{2}}\tilde{\bar{\mathbf{Q}}}A^{-\frac{1}{2}}\mathbf{W}\right)}{TL} + \frac{\mathbf{W}^\top\mathbf{A}^{-\frac{1}{2}}\tilde{\bar{\mathbf{Q}}}_2(\mathbf{A})\mathbf{A}^{-\frac{1}{2}}\mathbf{W}}{TL} + \frac{1}{TL}\mathrm{tr}(\boldsymbol{\Sigma}_N\bar{\mathbf{Q}}_2) + O\left(\frac{1}{\sqrt{d}}\right)
\end{aligned}
$$

The test risk can be further simplified as

$$
\mathcal{R}^{\infty}_{test} = tr\left(\boldsymbol{\Sigma}_N\right) + \frac{\mathbf{W}^\top\mathbf{A}^{-\frac{1}{2}}\tilde{\bar{\mathbf{Q}}}_2(\mathbf{A})\mathbf{A}^{-\frac{1}{2}}\mathbf{W}}{TL} + \frac{tr\left(\boldsymbol{\Sigma}_N\bar{\mathbf{Q}}_2\right)}{TL} + O\left(\frac{1}{\sqrt{d}}\right)
$$

B.3.2 Train Risk

In this section, we derive the asymptotic risk for the training data.

Theorem B.3.1 (Asymptotic training risk). *Assuming that the training data vectors $\mathbf{x}_i^{(t)}$ and the test data vectors $\mathbf{x}^{(t)}$ are concentrated random vectors, and given the growth rate assumption (Assumption 4.2.2), it follows that:*

$$\mathcal{R}^{\infty}_{train} \leftrightarrow \frac{1}{Tn} tr\left(\mathbf{W}^\top \mathbf{A}^{-1/2} \bar{\tilde{\mathbf{Q}}} \mathbf{A}^{-1/2} \mathbf{W}\right) - \frac{1}{Tn} tr\left(\mathbf{W}^\top \mathbf{A}^{-1/2} \bar{\tilde{\mathbf{Q}}}_2(\mathbf{I}_{TL}) \mathbf{A}^{-1/2} \mathbf{W}\right) + \frac{1}{Tn} tr\left(\mathbf{\Sigma}_N \bar{\mathbf{Q}}_2\right)$$

Proof. We aim in this setting of regression, to compute the asymptotic theoretical training risk given by:

$$\mathcal{R}^{\infty}_{train} = \frac{1}{Tn} \mathbb{E}\left[\left\| \mathbf{Y} - \frac{\mathbf{Z}^\top \mathbf{A} \mathbf{Z}}{TL} \mathbf{Q} \mathbf{Y} \right\|_2^2\right]$$

Using the definition of $\mathbf{Q} = \left(\frac{\mathbf{Z}^\top \mathbf{A} \mathbf{Z}}{TL} + \mathbf{I}_{TL}\right)^{-1}$, we have that

$$\frac{\mathbf{Z}^\top \mathbf{A} \mathbf{Z}}{TL} \mathbf{Q} = \left(\frac{\mathbf{Z}^\top \mathbf{A} \mathbf{Z}}{TL} + \mathbf{I}_{TL} - \mathbf{I}_{TL}\right) \left(\frac{\mathbf{Z}^\top \mathbf{A} \mathbf{Z}}{TL} + \mathbf{I}_{TL}\right)^{-1} = \mathbf{I}_{TL} - \mathbf{Q}$$

Plugging back into the expression of the training risk then leads to

$$\mathcal{R}^{\infty}_{train} = \frac{1}{Tn} \mathbb{E}\left[tr\left(\mathbf{Y}^\top \mathbf{Q}^2 \mathbf{Y}\right)\right]$$

Using the definition of the linear generative model and in particular $\mathbf{Y} = \frac{\mathbf{Z}^\top \mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon}$, we get

$$\begin{aligned}
\mathcal{R}^{\infty}_{train} &= \frac{1}{Tn} \mathbb{E}\left[tr\left(\frac{1}{\sqrt{TL}} \mathbf{Z}^\top \mathbf{W} + \boldsymbol{\varepsilon}\right)^\top \mathbf{Q}^2 \left(\frac{1}{\sqrt{TL}} \mathbf{Z}^\top \mathbf{W} + \boldsymbol{\varepsilon}\right)\right] \\
&= \frac{1}{Tn} \frac{1}{TL} \mathbb{E}\left[tr\left(\mathbf{W}^\top \mathbf{Z} \mathbf{Q}^2 \mathbf{Z}^\top \mathbf{W}\right)\right] + \frac{1}{Tn} \mathbb{E}\left[tr\left(\boldsymbol{\varepsilon}^\top \mathbf{Q}^2 \boldsymbol{\varepsilon}\right)\right]
\end{aligned}$$

To simplify this expression, we will introduced the so-called "coresolvent" defined as:

$$\tilde{\mathbf{Q}} = \left(\frac{\mathbf{A}^{\frac{1}{2}} \mathbf{Z} \mathbf{Z}^\top \mathbf{A}^{\frac{1}{2}}}{TL} + \mathbf{I}_{TL}\right)^{-1},$$

Employing the elementary relation $\mathbf{A}^{\frac{1}{2}} \mathbf{Z} \mathbf{Q} = \tilde{\mathbf{Q}} \mathbf{A}^{\frac{1}{2}} \mathbf{Z}$, one obtains:

$$\frac{1}{TL} \mathbf{Z} \mathbf{Q}^2 \mathbf{Z}^\top = \frac{1}{TL} \mathbf{A}^{-\frac{1}{2}} \tilde{\mathbf{Q}} \mathbf{A}^{\frac{1}{2}} \mathbf{Z} \mathbf{Q} \mathbf{Z}^\top = \mathbf{A}^{-\frac{1}{2}} \tilde{\mathbf{Q}}^2 \frac{\mathbf{A}^{\frac{1}{2}} \mathbf{Z} \mathbf{Z}^\top \mathbf{A}^{\frac{1}{2}}}{TL} \mathbf{A}^{-\frac{1}{2}} = \mathbf{A}^{-\frac{1}{2}} \tilde{\mathbf{Q}} \mathbf{A}^{-\frac{1}{2}} - \mathbf{A}^{-\frac{1}{2}} \tilde{\mathbf{Q}}^2 \mathbf{A}^{-\frac{1}{2}},$$

Therefore we further get

$$\mathcal{R}^{\infty}_{train} = \frac{1}{Tn} \mathbb{E}\left[tr\left(\mathbf{W}^\top \mathbf{A}^{-1/2} \tilde{\mathbf{Q}} \mathbf{A}^{-1/2} \mathbf{W}\right)\right] - \frac{1}{Tn} \mathbb{E}\left[tr\left(\mathbf{W}^\top \mathbf{A}^{-1/2} \tilde{\mathbf{Q}}^2 \mathbf{A}^{-1/2} \mathbf{W}\right)\right] + \frac{1}{Tn} \mathbb{E}\left[tr\left(\boldsymbol{\varepsilon}^\top \mathbf{Q}^2 \boldsymbol{\varepsilon}\right)\right]$$

Using deterministic equivalents in Lemma 1, the training risk then leads to

$$\mathcal{R}^{\infty}_{train} = \frac{1}{Tn} tr\left(\mathbf{W}^\top \mathbf{A}^{-1/2} \bar{\tilde{\mathbf{Q}}} \mathbf{A}^{-1/2} \mathbf{W}\right) - \frac{1}{Tn} tr\left(\mathbf{W}^\top \mathbf{A}^{-1/2} \bar{\tilde{\mathbf{Q}}}_2(\mathbf{I}_{TL}) \mathbf{A}^{-1/2} \mathbf{W}\right) + \frac{1}{Tn} tr\left(\mathbf{\Sigma}_N \bar{\mathbf{Q}}_2\right) + O\left(\frac{1}{\sqrt{L}}\right)$$

□

B.4 Interpretation and insights of the theoretical analysis

B.4.1 Analysis of the test risk

We recall the test risk as

$$\mathcal{R}_{test}^{\infty} = tr\left(\mathbf{\Sigma}_N\right) + \frac{\mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}} \bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}) \mathbf{A}^{-\frac{1}{2}} \mathbf{W}}{TL} + \frac{tr\left(\mathbf{\Sigma}_N \bar{\mathbf{Q}}_2\right)}{TL} + O\left(\frac{1}{\sqrt{L}}\right)$$

The test risk is composed of a signal term of a signal term $\mathcal{S} = \frac{\mathrm{W}^\top \mathrm{A}^{-\frac{1}{2}} \bar{\tilde{\mathrm{Q}}}_2(\mathrm{A}) \mathrm{A}^{-\frac{1}{2}} \mathrm{W}}{TL}$ and a noise term $\mathcal{N} = \frac{tr\left(\Sigma_N \bar{\mathrm{Q}}_2\right)}{TL}$.

B.4.2 Interpretation of the signal term

Let's denote by $\bar{\mathbf{\Sigma}} = \sum_{t=1}^{T} \frac{n_T L_t}{TL(1+\delta_t)^2} \mathbf{\Sigma}^{(t)}$ and $\tilde{\mathbf{\Sigma}} = \sum_{t=1}^{T} \frac{c_0}{1+\delta_t} \mathbf{\Sigma}^{(t)}$. The signal term reads as

$$\mathcal{S} = \mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}} \bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}) \mathbf{A}^{-\frac{1}{2}} \mathbf{W}.$$

Using the following identity,

$$\begin{aligned} \mathbf{A}^{-\frac{1}{2}} \bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}) \mathbf{A}^{-\frac{1}{2}} &= \mathbf{A}^{-\frac{1}{2}} \bar{\mathbf{Q}} \mathbf{A}^{\frac{1}{2}} \left(\mathbf{I} + \bar{\mathbf{\Sigma}}\right) \mathbf{A}^{\frac{1}{2}} \bar{\mathbf{Q}} \mathbf{A}^{-\frac{1}{2}} \\ &= \left(\mathbf{A}\tilde{\mathbf{\Sigma}} + \mathbf{I}\right)^{-1} \left(\mathbf{I} + \bar{\mathbf{\Sigma}}\right) \left(\mathbf{A}\tilde{\mathbf{\Sigma}} + \mathbf{I}\right)^{-1} \end{aligned}$$

This finally leads to

$$\mathcal{S} = \mathbf{W}^\top \left(\mathbf{A}\tilde{\mathbf{\Sigma}} + \mathbf{I}\right)^{-1} \left(\mathbf{I} + \bar{\mathbf{\Sigma}}\right) \left(\mathbf{A}\tilde{\mathbf{\Sigma}} + \mathbf{I}\right)^{-1} \mathbf{W}$$

The matrix $\mathcal{H} = \left(\mathbf{A}\tilde{\mathbf{\Sigma}} + \mathbf{I}\right)^{-1} \left(\mathbf{I} + \bar{\mathbf{\Sigma}}\right) \left(\mathbf{A}\tilde{\mathbf{\Sigma}} + \mathbf{I}\right)^{-1}$ is responsible to amplifying the signal $\mathbf{W}^\top \mathbf{W}$ in order to let the test risk to decrease more or less. It is is decreasing as function of the number of samples in the tasks n_t. Furthermore it is composed of two terms (from the independent training $\mathbf{W}_t^\top \mathbf{W}$) and the cross term $\mathbf{W}_t^\top \mathbf{W}_v$ for $t \neq v$. Both terms decreases as function of the number of samples n_t, smaller values of γ_t and increasing value of λ. The cross term depends on the matrix $\mathbf{\Sigma}_t^{-1} \mathbf{\Sigma}_v$ which materializes the covariate shift between the tasks. More specifically, if the features are aligned $\mathbf{\Sigma}_t^{-1} \mathbf{\Sigma}_v = I$ and the cross term is maximal while for bigger Fisher distance between the covariance of the tasks, the correlation is not favorable for multi task learning. To be more specific the off-diagonal term of $\mathcal{H}$ are responsible for the cross term therefore for the multi tasks and the diagonal elements are responsible for the independent terms.

To analyze more the element of $\mathcal{H}$, let's consider the case where $\Sigma^{(t)} = \mathbf{I}$ and $\gamma_t = \gamma$. In this case the diagonal and non diagonal elements $\mathbf{D}_{IL}$ and $\mathbf{C}_{MTL}$ are respectively given by

$$\mathbf{D}_{IL} = \frac{(c_0(\lambda+\gamma)+1)^2 + c_0^2\lambda^2}{(c_0(\lambda+\gamma)+1)^2 - c_0^2\lambda^2}, \quad \mathbf{C}_{MTL} = \frac{-2c_0\lambda(c_0(\lambda+\gamma)+1)}{(c_0(\lambda+\gamma)+1)^2 - c_0^2\lambda^2}$$

Both function are decreasing function of λ, $1/\gamma$ and c_0.

B.4.3 Interpretation and insights of the noise terms

We recall the definition of the noise term $\mathcal{N}$ as

$$\mathcal{N} = tr\left(\mathbf{\Sigma}_N \left(\mathbf{A}^{-1} + \mathbf{\Sigma}\right)^{-1}\right)$$

Now at the difference of the signal term there are no cross terms due to the independence between the noise of the different tasks. In this case on the diagonal elements of $(\mathbf{A}^{-1} + \mathbf{\Sigma})^{-1}$ matters. This diagonal term is increasing for an increasing value of the sample size, the value of λ. Therefore this term is responsible for the negative transfer. In the specific case where $\mathbf{\Sigma}^{(t)} = \mathbf{I}_d$ and $\gamma_t = \gamma$ for all task t, the diagonal terms read as

$$\mathbf{N}_{NT} = \frac{(c_0(\lambda+\gamma)^2 + (\lambda+\gamma) - c_0\lambda^2)^2 + \lambda^2}{((c_0(\lambda+\gamma)+1)^2 - c_0^2\lambda^2)^2}$$

B.4.4 Optimal Lambda

The test risk in the particular of identity covariance matrix can be rewritten as

$$\mathcal{R}^{\infty}_{test} = \mathbf{D}_{IL}\left(\|\mathbf{W}_1\|_2^2 + \|\mathbf{W}_2\|_2^2\right) + \mathbf{C}_{MTL}\mathbf{W}_1^\top\mathbf{W}_2 + \mathbf{N}_{NT}\,\mathrm{Tr}(\mathbf{\Sigma})_n.$$

Deriving $\mathcal{R}^{\infty}_{test}$ with respect to λ leads after some algebraic calculus to

$$\lambda^\star = \frac{n}{L}SNR - \frac{\gamma}{2}$$

where the signal noise ratio is composed of the independent signal to noise ratio and the cross signal to noise ratio $SNR = \frac{\|W_1\|_2^2 + W_2\|_2^2}{\mathrm{tr}\left(\Sigma_n\right)} + \frac{W_1^\top W_2}{\mathrm{tr}\left(\Sigma_n\right)}$

B.5 Theoretical Estimations

B.5.1 Estimation of the training and test risk

The different theorems depends on the ground truth $\mathbf{W}$ that needs to be estimated through $\hat{\mathbf{W}}$.

To estimate the test risk, one needs to estimate functionals of the form $\mathbf{W}^\top \mathbf{M}\hat{\mathbf{W}}$ and $\boldsymbol{\varepsilon}^\top \mathbf{M}\boldsymbol{\varepsilon}$ for any matrix $\mathbf{M}$. Using the expression of $\mathbf{W} = \mathbf{AZQY}$, we start computing $\hat{\mathbf{W}}^\top \mathbf{M}\hat{\mathbf{W}}$

$$\hat{\mathbf{W}}^\top \mathbf{MW} = \mathbf{Y}^\top \mathbf{QZ}^\top \mathbf{AMAZQY}$$

Using the generative model for $\mathbf{Y} = \frac{\mathbf{Z}^\top \mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon}$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\hat{\mathbf{W}}^\top \mathbf{MW}\right] &= \mathbb{E}\left[\left(\frac{\mathbf{Z}^\top \mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon}\right)^\top \mathbf{QZ}^\top \mathbf{AMAZQ}\left(\frac{\mathbf{Z}^\top \mathbf{W}}{\sqrt{TL}} + \boldsymbol{\varepsilon}\right)\right] \\
&= \frac{1}{TL}\mathbb{E}\left[\mathbf{W}^\top \mathbf{ZQZ}^\top \mathbf{AMAZQZ}^\top \mathbf{W}\right] + \mathbb{E}\left[\boldsymbol{\varepsilon}^\top \mathbf{QZ}^\top \mathbf{AMAZQ}\boldsymbol{\varepsilon}\right]
\end{aligned}$$

Employing the elementary relation $\mathbf{A}^{\frac{1}{2}}\mathbf{ZQ} = \tilde{\mathbf{Q}}\mathbf{A}^{\frac{1}{2}}\mathbf{Z}$, one obtains:

$$\begin{aligned}
\mathbb{E}\left[\hat{\mathbf{W}}^\top \mathbf{MW}\right] &= \frac{1}{TL}\mathbb{E}\left[\mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}}\tilde{\mathbf{Q}}\mathbf{A}^{\frac{1}{2}}\mathbf{Z}^\top \mathbf{ZA}^{\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}}\tilde{\mathbf{Q}}\mathbf{A}^{\frac{1}{2}}\mathbf{ZZ}^\top \mathbf{W}\right] + \mathbb{E}\left[\boldsymbol{\varepsilon}^\top \mathbf{QZ}^\top \mathbf{AMAZQ}\boldsymbol{\varepsilon}\right] \\
&= \mathbb{E}\left[\mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}}\left(\mathbf{I} - \tilde{\mathbf{Q}}\right)\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}}\left(\mathbf{I} - \tilde{\mathbf{Q}}\right)\mathbf{A}^{-\frac{1}{2}}\mathbf{W}\right] + \mathbb{E}\left[\boldsymbol{\varepsilon}^\top \mathbf{QZ}^\top \mathbf{AMAZQ}\boldsymbol{\varepsilon}\right] \\
&= \mathbb{E}\left[\mathbf{W}^\top \mathbf{MW}\right] - 2\mathbb{E}\left[\mathbf{W}^\top \mathbf{MA}^{\frac{1}{2}}\tilde{\mathbf{Q}}\mathbf{A}^{-\frac{1}{2}}\mathbf{W}\right] + \mathbb{E}\left[\mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}}\tilde{\mathbf{Q}}\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}}\tilde{\mathbf{Q}}\mathbf{A}^{-\frac{1}{2}}\mathbf{W}\right] \\
&\quad + \mathbb{E}\left[\boldsymbol{\varepsilon}^\top \mathbf{QZ}^\top \mathbf{AMAZQ}\boldsymbol{\varepsilon}\right]
\end{aligned}$$

Using the deterministic equivalent of Lemma 1, we obtain

$$\begin{aligned}
\hat{\mathbf{W}}^\top \mathbf{M}\hat{\mathbf{W}} &\leftrightarrow \mathbf{W}^\top \mathbf{MW} - 2\mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}\mathbf{A}^{\frac{1}{2}}\mathbf{MW} + \mathrm{Tr}(\boldsymbol{\Sigma})_n \mathbf{M}(\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}}) + \mathbf{W}^\top \mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}})\mathbf{A}^{-\frac{1}{2}}\mathbf{W} \\
&\leftrightarrow \mathbf{W}^\top \left(\mathbf{M} - 2\mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}\mathbf{A}^{\frac{1}{2}}\mathbf{M} + \mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}})\mathbf{A}^{-\frac{1}{2}}\right)\mathbf{W} + \mathrm{Tr}(\boldsymbol{\Sigma})_n \bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}}) \\
&\leftrightarrow \mathbf{W}^\top \kappa(\mathbf{M})\mathbf{W} + \mathrm{Tr}(\boldsymbol{\Sigma})_n \bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}})
\end{aligned}$$

where We define the mapping $\kappa : \mathbb{R}^{TL\times TL} \to \mathbb{R}^{H\times H}$ as follows

$$\kappa(\mathbf{M}) = \mathbf{M} - 2\mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}\mathbf{A}^{\frac{1}{2}}\mathbf{M} + \mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{MA}^{\frac{1}{2}})\mathbf{A}^{-\frac{1}{2}}.$$

B.5.2 Estimation of the noise covariance

The estimation of the noise covariance remains a technical challenge in this process. However, when the noise covariance is isotropic, it is sufficient to estimate only the noise variance. By observing that

$$\lim_{\lambda\to 0,\gamma\to\infty} \mathcal{R}^{\infty}_{train} = \sigma^2 \frac{\mathrm{Tr}(\mathbf{Q})_2}{kn},$$

we can estimate the noise level from the training risk evaluated at large γ and $\lambda = 0$.

B.5.3 Empirical Estimation of Task-wise Signal, Cross Signal, and Noise

All the quantities defined in Theorem 4.3.1 are known except for the bilinear expressions $\frac{1}{TL}tr(\mathbf{W}^\top\mathbf{M}\mathbf{W})$ and $\frac{1}{TL}tr\left(\mathbf{\Sigma}_N\mathbf{M}\right)$. These quantities can be consistently estimated under Assumptions 4.2.2 as follows :

$$\frac{1}{TL}tr\left(\mathbf{W}^\top\mathbf{M}\mathbf{W}\right)-\zeta(\mathbf{M})\xrightarrow{\text{a.s.}}0,\qquad \frac{1}{TL}tr\left(\mathbf{\Sigma}_N\mathbf{M}\right)-\hat{\sigma}tr\left(\mathbf{M}\right)\xrightarrow{\text{a.s.}}0$$

where $\zeta(\mathbf{M})=\frac{1}{TL}tr(\hat{\mathbf{W}}^\top\kappa^{-1}(\mathbf{M})\mathbf{M}\hat{\mathbf{W}})-\frac{\hat{\sigma}}{TL}tr\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\kappa^{-1}(\mathbf{M})\mathbf{A}^{\frac{1}{2}})$.

We define the estimate of the noise as $\hat{\sigma}=\lim\limits_{\substack{\lambda\to 0\\ \gamma_t\to\infty}}\mathcal{R}^{\infty}_{train}$ and the function κ^{-1} is the functional inverse of the mapping $\kappa:\mathbb{R}^{TL\times TL}\to\mathbb{R}^{H\times H}$ defined as follows:

$$\kappa(\mathbf{M})=\mathbf{M}-2\mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}\mathbf{A}^{\frac{1}{2}}\mathbf{M}+\mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{M}\mathbf{A}^{\frac{1}{2}})\mathbf{A}^{-\frac{1}{2}}-\frac{n}{(TL)^2}\mathbf{A}^{-\frac{1}{2}}\mathbf{\Sigma}\mathbf{A}^{-\frac{1}{2}}\bar{\tilde{\mathbf{Q}}}_2(\mathbf{A}^{\frac{1}{2}}\mathbf{M}\mathbf{A}^{\frac{1}{2}}).$$

B.6 Application to Multi-task Regression

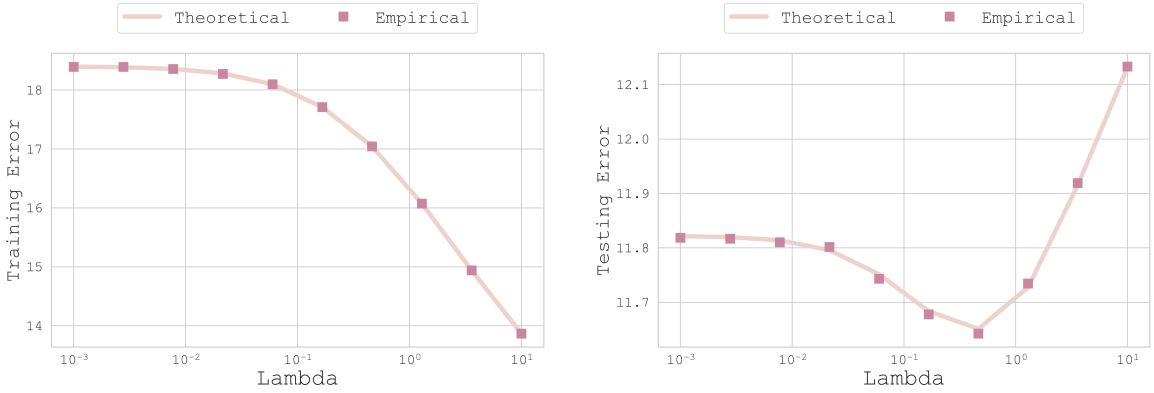
B.6.1 Related Work

High-Dimensional Regression Analysis. High-dimensional regression has been extensively studied in single-task settings using RMT Dobriban & Wager, 2018 and other statistical methods Gerbelot et al., 2022. These works typically focus on linear signal-plus-noise models to derive test risk based on signal parameters and noise covariance. Our research extends these concepts to MTL, providing unique insights into the effects of shared and task-specific learning. Unlike previous studies, we offer a practical approach to estimate asymptotic test risks and optimize hyperparameters, making our theoretical findings actionable within the MTL framework for multivariate forecasting.

B.6.2 Empirical vs. Theoretical Comparison

In our study, we apply the theoretical framework presented in our paper to a real-world regression problem, specifically, the *Appliance Energy dataset* which aims to predict the total usage of a house. This dataset is a multivariate regression dataset containing 138 time series, each of dimension 24. We select two of these features as 2 tasks to cast the problem as a multi-task learning regression problem.

Figure B.1 presents a comparison between the theoretical predictions and empirical outcomes of our proposed linear framework. Despite the assumptions made in the main body of the paper, the theoretical predictions closely align with the experimental results.



(a) Training MSE

(b) Testing MSE

Figure B.1: Theoretical vs Empirical MSE as function of regularization parameter λ. Close fit between the theoretical and the empirical predictions which underscores the robustness of the theory in light of varying assumptions as well as the accuracy of the suggested estimates. We consider the first two channels as the the two tasks and $L = 144$. 95 samples are used for the training and 42 samples are used for the test.

This demonstrates that our estimates are effective in practice and provide a reliable approximation of the optimal regularization.

In essence, our theoretical framework, when applied to real-world multi-task learning regression problems, yields practical and accurate estimates, thereby validating its effectiveness and applicability.

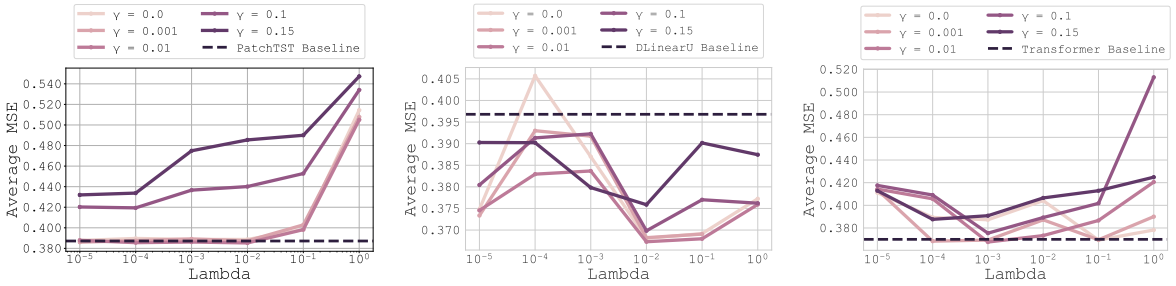
B.7 Additional Experiments

In this section, we present the results of our multi-task learning regularization framework on the dataset ETTh1.

B.8 Limitations

While the study provides valuable insights through its theoretical analysis within a linear framework, it is important to acknowledge its limitations. The linear approach serves as a solid foundation for understanding more complex models, but its practical applications may be constrained. Linear models, though mathematically tractable and often easier to interpret, might not fully capture the intricacies and nonlinear relationships present in real-world data, especially in the context of multivariate time series forecasting.

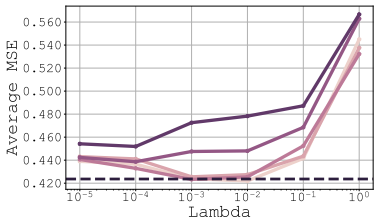
To address this limitation, we decided to extend our algorithm's application to more complex models, specifically within the nonlinear setting of neural networks. This transi-



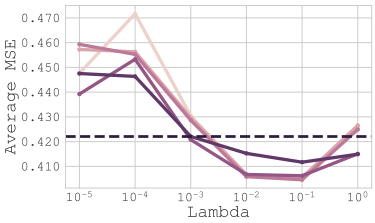
(a) PatchTST - Horizon 96

(b) DLinearU - Horizon 96

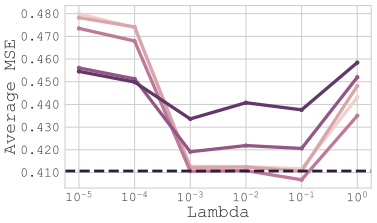
(c) Transformer - Horizon 96



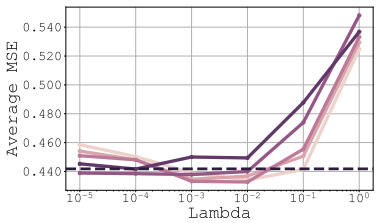
(d) PatchTST - Horizon 192



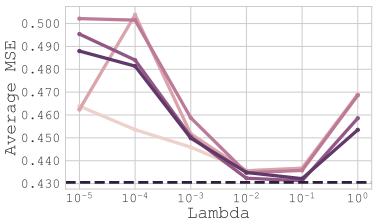
(e) DLinearU - Horizon 192



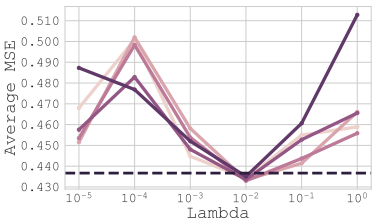
(f) Transformer - Horizon 192



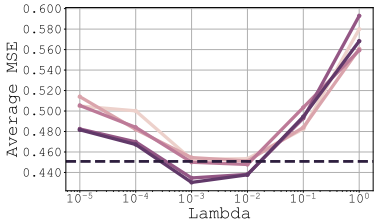
(g) PatchTST - Horizon 336



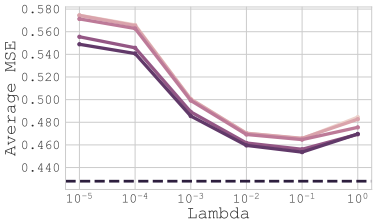
(h) DLinearU - Horizon 336



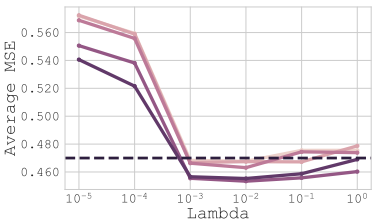
(i) Transformer - Horizon 336



(j) PatchTST - Horizon 720



(k) DLinearU - Horizon 720



(l) Transformer - Horizon 720

Figure B.2: Results for dataset ETTh1 on the PatchTST, DLinearU, and Transformer baselines, averaged across 3 seeds for each gamma and lambda setting.

tion aims to evaluate whether the theoretical insights derived from the linear framework hold true empirically when applied to neural networks. As part of this endeavor, an optimal parameter lambda was selected by an oracle, leading to promising results, as detailed in Section 4.4.2. This oracle-based selection underscores the potential efficacy of our approach when appropriately tuned, even in more complex, nonlinear contexts.

It is important to note that the limitations are not related to the real-world data itself, as our setting performs well in the context of multi-task regression for real-world data, as shown in Section B.6. The difficulty arises from transitioning from a linear to a nonlinear model. The results in Section 4.4.2 are particularly encouraging, demonstrating that our method can improve upon univariate baselines by regularizing with an optimal lambda, as indicated by our oracle. While the oracle provides an upper bound on performance, actual implementation would require robust methods for hyperparameter optimization in non-linear scenarios, which remains an open area for further research.

By expanding the scope of our theoretical framework to encompass nonlinear models, we pave the way for future work that could focus on the theoretical analysis of increasingly complex architectures.