SAMformer : Unlocking the potential of Transformers in Time-Series Forecasting

Presentation : Romain ILBERT*

Check my website for code and paper :

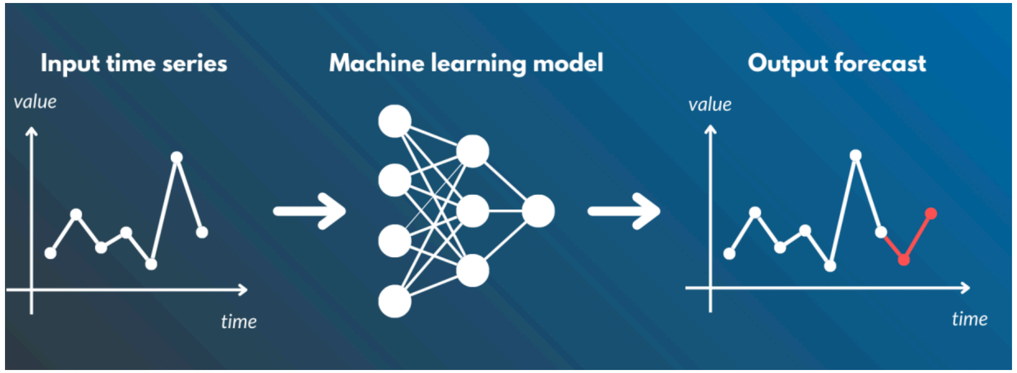
* PhD Student at LIPADE, Paris Descartes University & Huawei Paris Research Center

Accepted as an Oral at ICML 2024

Time Series Forecasting : A Definition

Problem Setup

1. Given past observations, predict future ones



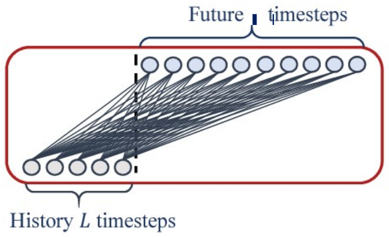
2. Univariate (single channel) vs. multivariate (multi-channels)

3. Short, medium and long-term horizon

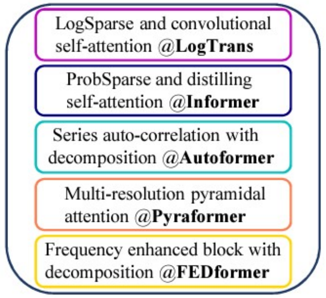
Failure of Transformers

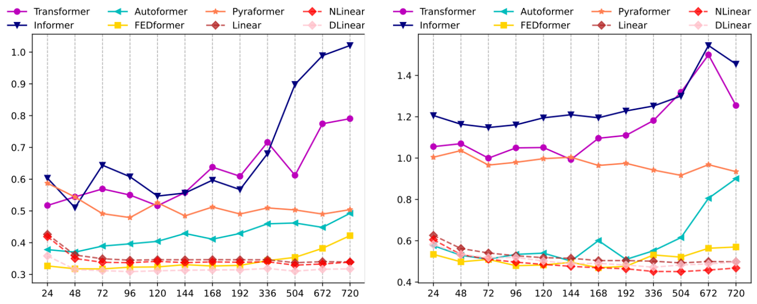
Motivation :

1. Consider the simplest linear model for forecasting
2. Compare it to state-of-the-art transformers



VS.





(a) **24** steps-**ETTh1** (b) **720** steps-**ETTh1**

This Linear model surpasses the SOTA FEDformer (ICML'22) in most cases by 20%~50%

Main conclusions from *Zeng et al.*

1. Transformer-based methods don't work well in forecasting
2. Embarrassing failure in most basic scenario

… yet they dominate NLP and vision. Why?

1. Are Transformers Effective for Time Series Forecasting ? Zeng et al. 2023.

SAMformer (Ilbert et al. , ICML Oral 2024)

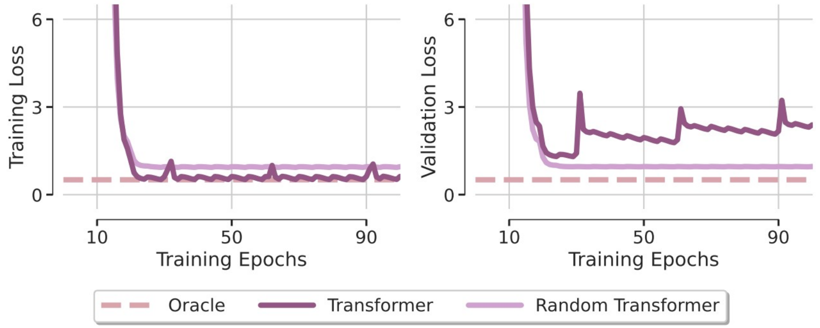
A transformer-based TS forecaster that actually works

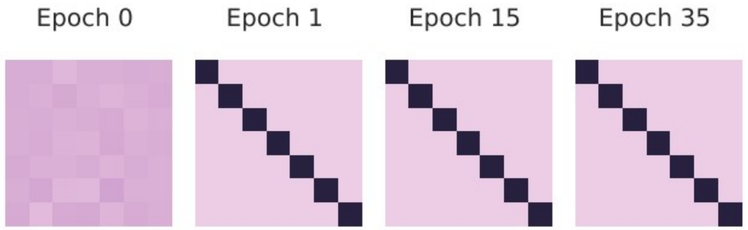
Simple Toy Regression Example

Context :

- Consider a simple linear regression problem
- Transformer defined as $f(\mathbf{X}) = [\mathbf{X} + A(\mathbf{X})\mathbf{X}\mathbf{W}_V\mathbf{W}_O]\mathbf{W}$ with **channel-wise attention** (DxD matrix, rather than LxL)

$$\mathbf{Y} = \mathbf{X}\mathbf{W}_{\text{toy}} + \epsilon$$ (L=512, H=96, D=7)





Conclusions :

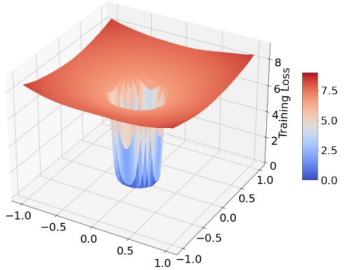
1. Linear Transformer severely overfits
2. And works better if we freeze the attention
3. Because the attention get stuck at the identity matrix and does not move afterwards

Pathological behavior suggesting sharp local minima

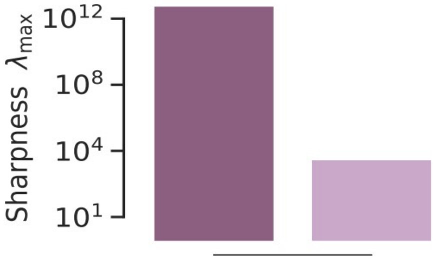
Simple Toy Regression Example

Why transformers fail?

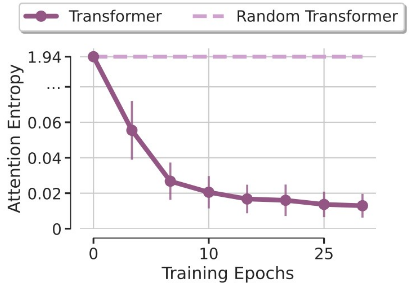
1. Transformers have a <u>sharp loss landscape</u> and suffer from <u>entropy collapse</u>



<u>High sharpness</u>

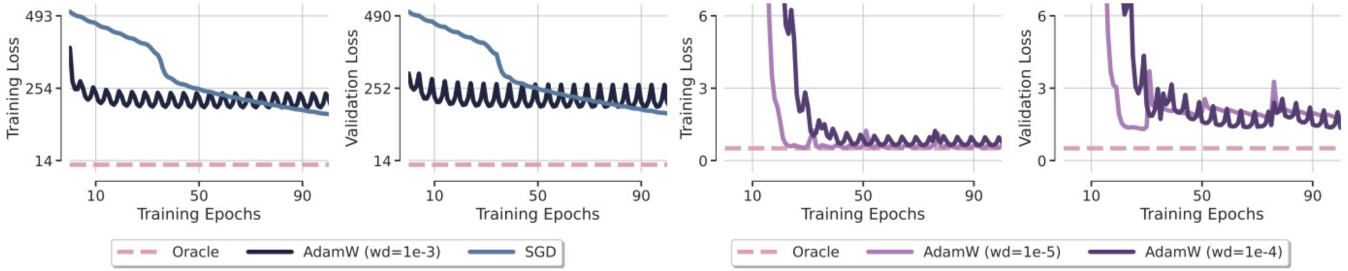


<u>Entropy collapse</u>



2. Well-known in NLP and vision (Chen et al., 2022, Zhai et al. 2023), ignored in TS Forecasting

3. And no changing in the optimizer helps to solve this



(a) SGD and AdamW with wd = 1e−3 (b) AdamW with wd ∈ {1e−5, 1e−4}.

Simple Toy Regression Example : To fix the sharpness issue

1. $\sigma -$ **Reparametrization** (Zhen et al. 2023)

   - Make attention matrix more uniform to avoid entropy collapse

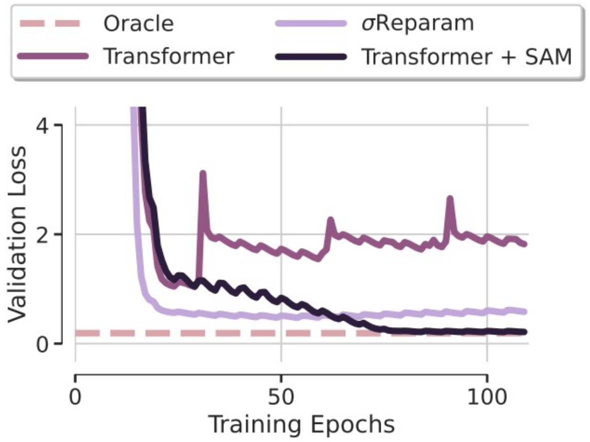
$$\hat{\mathbf{W}} = \frac{\gamma}{\|\mathbf{W}\|_2}\mathbf{W}$$

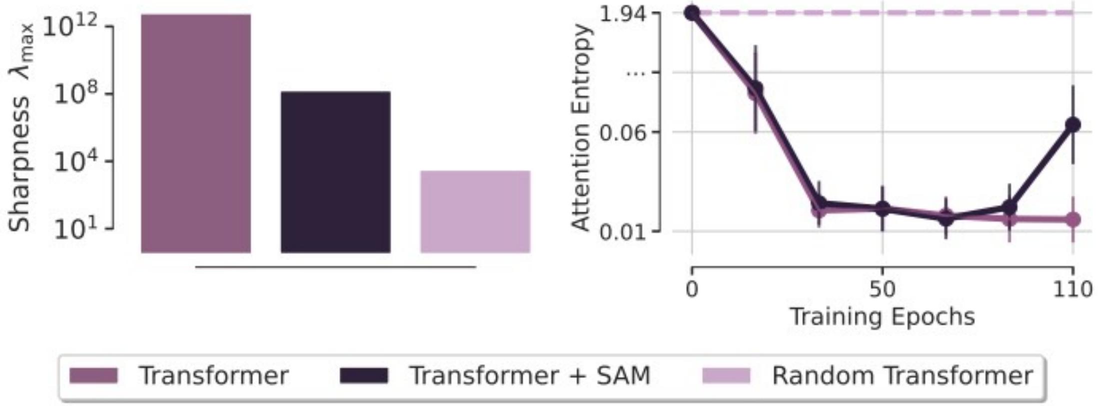
2. **Sharpness - Aware Minimization** (Foret et al. 2021, Chen et al. 2022)

   - Converge toward weights that lie in neighborhoods having uniformly low loss

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

SAM = desired solution





Congrats you now know how to solve

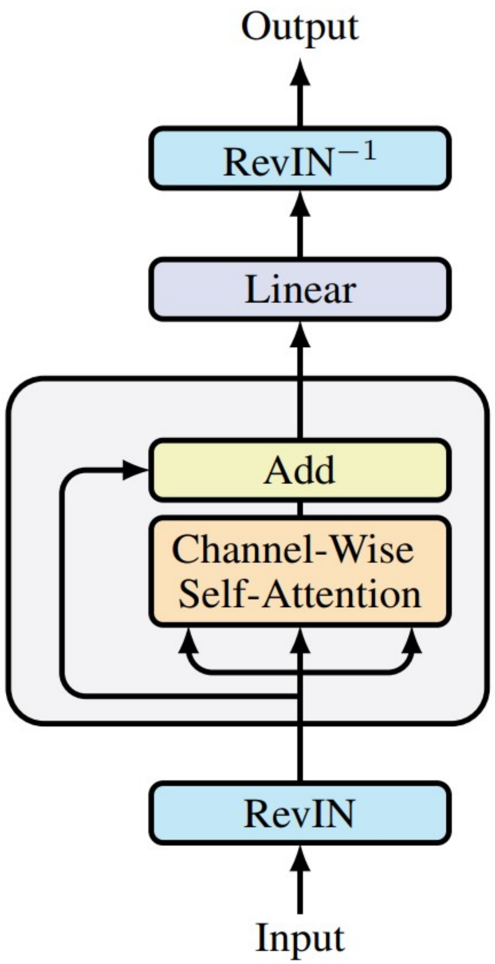
a linear regression problem with transformers!

SAMformer Architecture (ILBERT et al, ICML oral 2024)

Let's put it all together now:

1. Shallow transformer with a **channel-wise attention**
2. **RevIN layer** to be robust to train/test time shift
3. We optimize it with **SAM**

SAMformer = 15 lines of code



Experimental results (ILBERT et al, ICML oral 2024)

1. Datasets

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

2. Baselines

- TSmixer: MLPmixer model from Google (SOTA in 2023)
- Transformers: iTransformer (ICLR'24), PatchTST (ICLR'23), FEDformer (ICML'22), Pyraformer (ICLR'22), Informer (AAAI'21), Autoformer (NeurIPS'21), LogTrans (NeurIPS'19)

3. Nbr of parameters

- SAMFormer is **smaller** and **more consistent** than TSMixer. The **same model** for all datasets/horizons
- Avg Ratio = nbre params TSMixer / nbre params SAMFormer

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**

Experimental results (ILBERT et al, ICML oral 2024)

	with SAM		without SAM						
	SAMformer	TSMixer	Transformer	TSMixer	In*	Auto*	FED*	Pyra†	LogTrans†
Overall MSE improvement		5.25%	16.96%	14.33%	72.20%	22.65%	12.36%	61.88%	70.88%

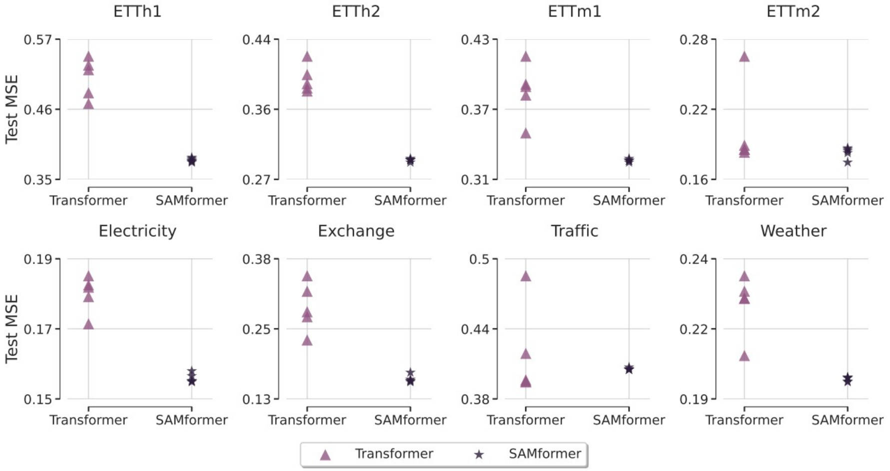
1. SAMformer is **14% better** than TSMixer, **11.13% better** than PatchTST, **3.94%** better than iTransformer

   - much better than all transformer-based models

2. <u>Sharpness-aware minimization</u> **improves** TSMixer as well

3. SAMformer is robust to random initialization

   - Very low variance for different random seeds compared to Transformer



Experimental results (ILBERT et al, ICML oral 2024)

SAMFormer is **on par with MOIRAI foundation model**

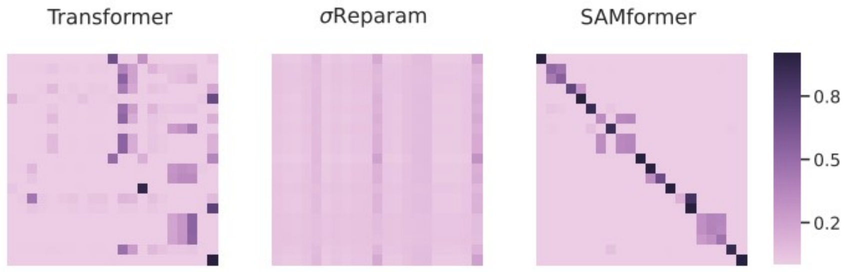
- MORAI (Salesforce + Singapore University)
- trained on LOTSA with 27B samples from 9 domains
- comes in 3 sizes: small (14M), base (91M) and Large (311M)

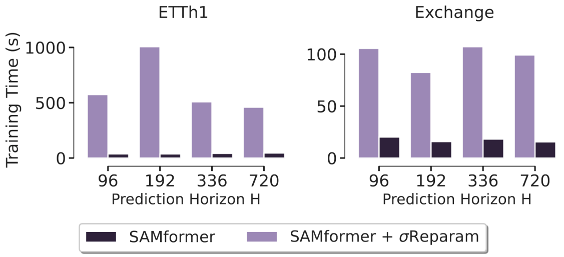
		$\text{MOIRAI}_{\text{Small}}$	$\text{MOIRAI}_{\text{Base}}$	$\text{MOIRAI}_{\text{Large}}$	SAMformer
ETTh1	MSE	**0.400**	0.434	0.510	0.41
	MAE	**0.424**	0.438	0.469	
ETTh2	MSE	**0.341**	0.345	0.354	0.344
	MAE	0.379	0.382	**0.376**	
ETTm1	MSE	0.448	**0.381**	0.390	**0.373**
	MAE	0.409	**0.388**	0.389	
ETTm2	MSE	0.300	**0.272**	0.276	**0.2685**
	MAE	0.341	0.321	**0.320**	
Electricity	MSE	0.233	0.188	0.188	**0.181**
	MAE	0.320	0.274	0.273	
Weather	MSE	0.242	**0.238**	0.259	0.26
	MAE	0.267	**0.261**	0.275	

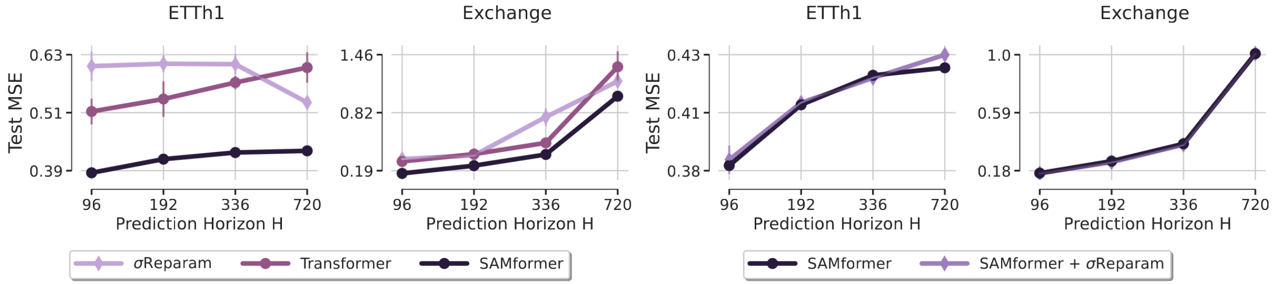
SAM vs σ – Reparametrization

Comparison with σ – **Reparametrization**

- proved to be efficient in NLP … but didn't work for us







(a) Comparison of `Transformer`, σReparam and `SAMformer`. (b) Comparison of `SAMformer` and `SAMformer` + σReparam.

Observations:

- Transformers ignores diagonal elements
- SAMformer strongly encourages feature self-correlation (as in ViTs)
- Weight reparametrization oversmoothes the attention matrix

SAM vs σ – Reparametrization

<u>Oversmoothing = rank collapse</u>

- we prove that

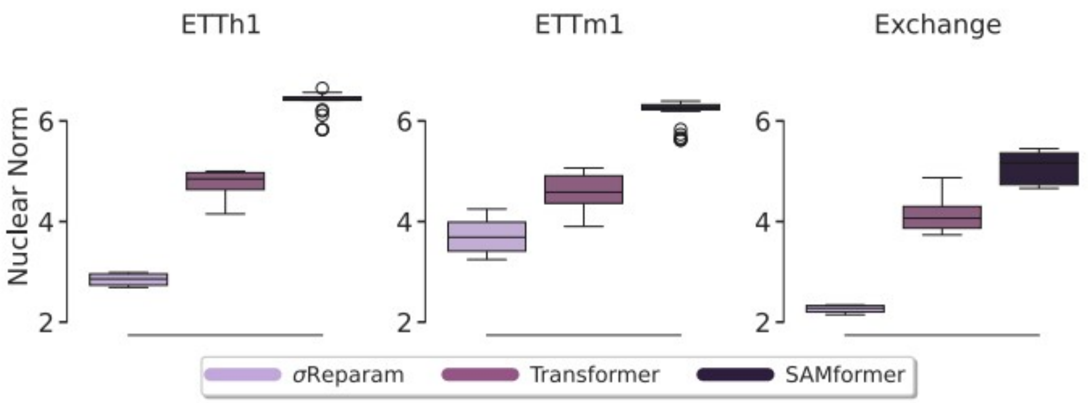
Proposition 2.2 (Upper bound on the nuclear norm) *Let* $\mathbf{X} \in \mathbb{R}^{D\times L}$ *be an input sequence. Assuming* $\mathbf{W}_Q\mathbf{W}_K^\top = \mathbf{W}_K\mathbf{W}_Q^\top \succcurlyeq \mathbf{0}$, *we have*

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

Roughly = **rank** of the **attention matrix**

Minimized by reparametrization

- maximizing the entropy of the attention = rank collapse
- rank collapse = uninformative channel-wise attention



Ablation Studies for SAMformer (ILBERT et al, ICML oral 2024)

Ablation study on channel-wise attention and identity weight matrix attention ?

- Candidate 1: SAMformer with **temporal** attention (as used in all other transformers)
- Candidate 2: SAMformer with **identity weight matrix** attention
- Overall Improvement : Improvement of SAMFormer over both candidates

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

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

Conclusions on SAMformer (ILBERT et al, ICML oral 2024)

1. We studied **pitfalls of transformers** in time series forecasting
   - Sharp loss landscape = lack of generalization
2. Our proposal **SAMformer**
   - **SAMformer =** RevIN + channel-wise attention + SAM optimization
   - **SOTA** in long-term multivariate time series forecasting
   - **Consistent** = same architecture of different horizons/datasets
   - **Lightweight** = the smallest SOTA model
   - On par with large foundation model MORAI

Thank you.