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

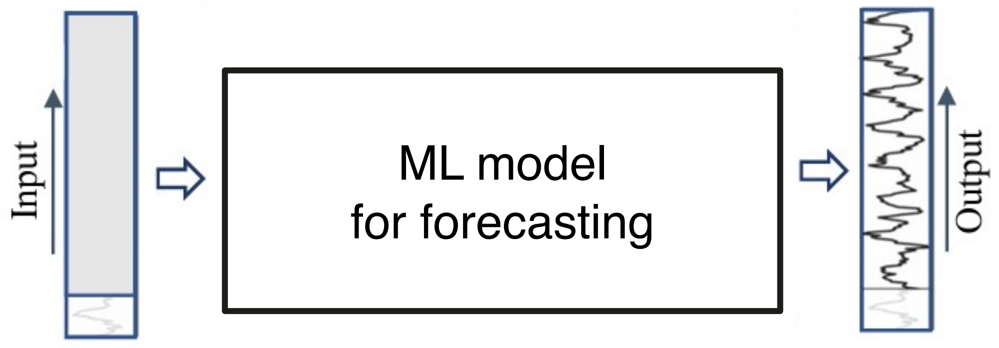
You can find this presentation on Romain ILBERT's website here : https://romilbert.github.io/

Accepted to ICML 2024

Introduction: Time Series Forecasting

Problem setup

1. Time series forecasting: given past observations , predict future ones



2. Univariate vs. multivariate (this work)

3. Short, medium and long-term (this work)

Introduction: Failure of Transformers

Motivation

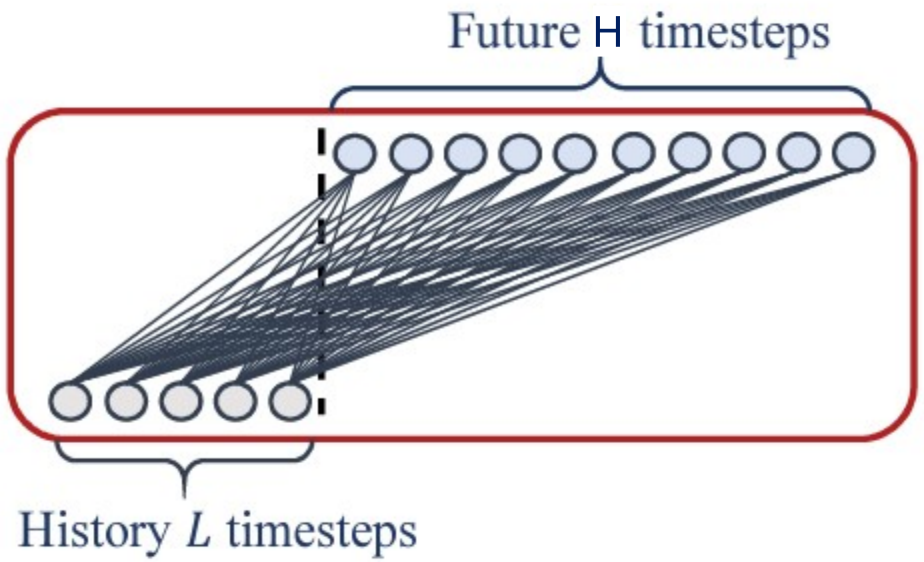
Are Transformers Effective for Time Series Forecasting?

Ailing Zeng[1*], Muxi Chen[1*], Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy (IDEA)
{alzeng, mxchen21, qxu}@cse.cuhk.edu.hk
{leizhang}@idea.edu.cn

Introduction: Failure of Transformers

Motivation

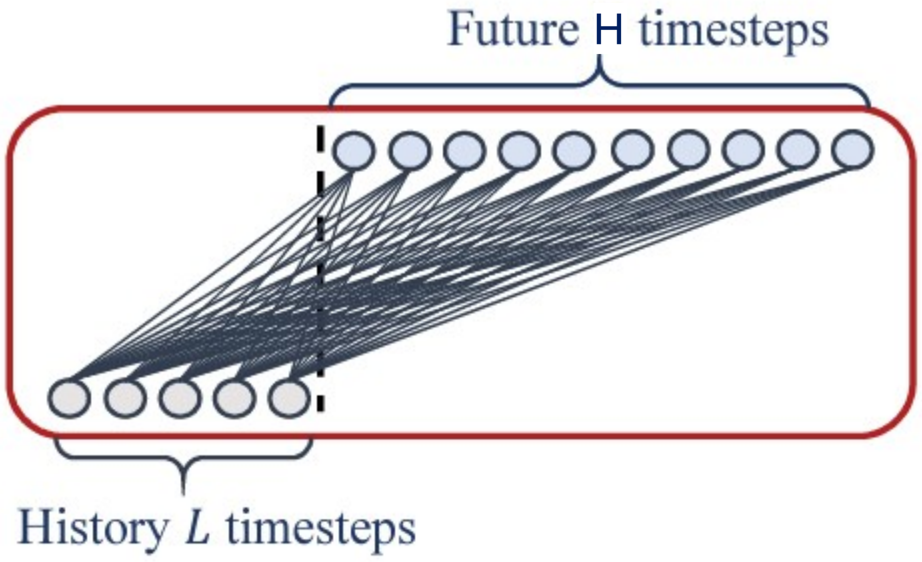
1. Consider a simple linear model per variable (no cross-feature correlations)



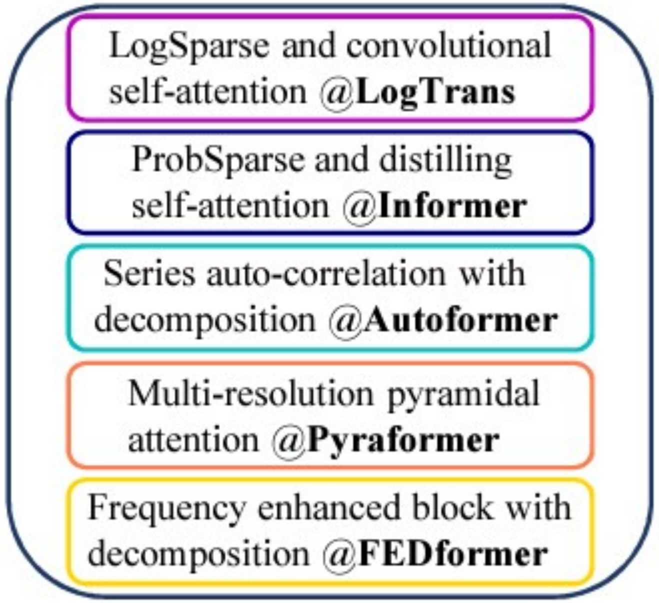
2. Compare it to « SOTA » transformers

Introduction: Failure of Transformers

Motivation



vs.



"Surprisingly, **Linear model surpasses the SOTA FEDformer** (ICML'22) in most cases by 20%~50%!"

Introduction: Failure of Transformers

Main conclusions by Zeng et al.

1. Existing **transformer**-based **methods don't work well** in forecasting
2. Embarrassing **failure** in most **basic scenarios**

… yet they <u>dominate NLP and vision</u>. Why?

SAMFormer (Ilbert et al. 2024)

A transformer-based forecaster that actually works

Introduction: basic example

Why transformers fail?

1. Consider a toy regression problem (L=512, H=96, D=7)

$$\mathbf{Y} = \mathbf{X}\mathbf{W}_{\text{toy}} + \varepsilon$$

2. Oracle = linear regression, closed-form solution

3. Competitor: shallow, linear transformer with **channel-wise attention** (DxD matrix, rather than LxL)

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

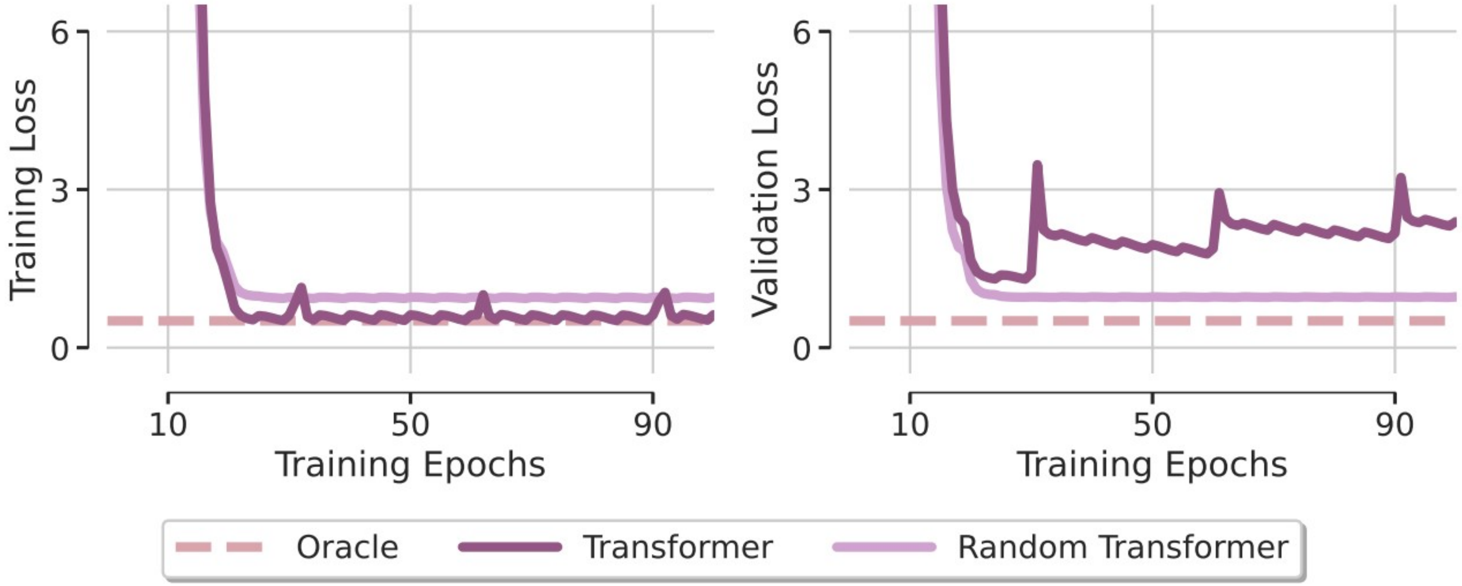
Can provably solve our problem!

HUAWEI

Introduction: basic example

Why transformers fail?

1. Linear, shallow transformer severly overfits!



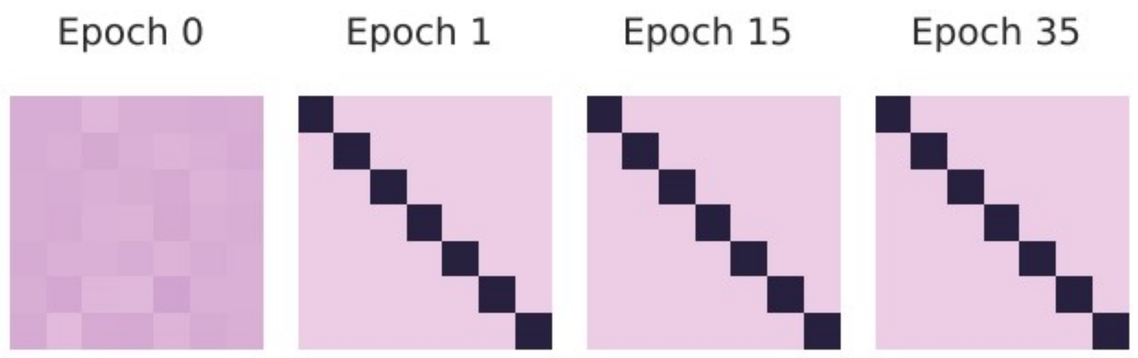
2. … but it **works better** if we **freeze the attention**

HUAWEI

Introduction: basic example

Why transformers fail?

1. Let’s look at the attention matrix



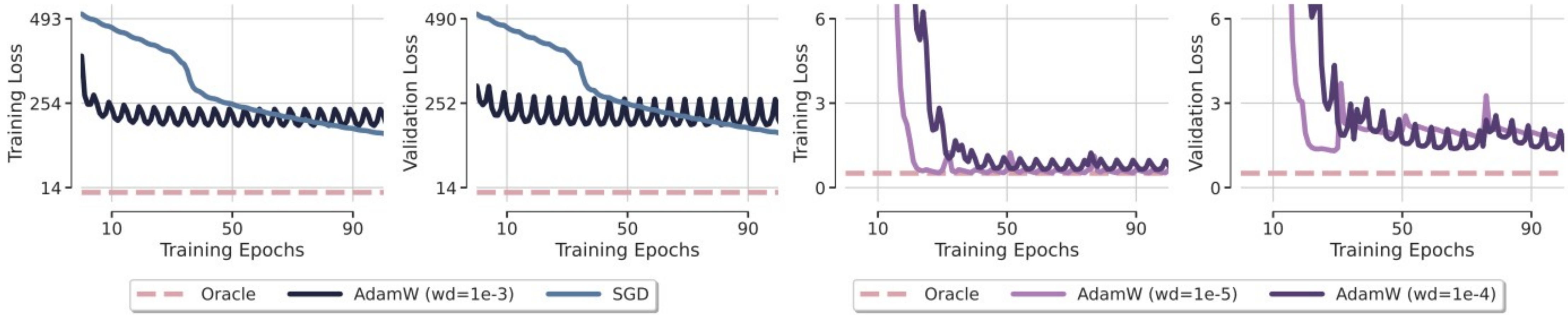
2. The attention get’s stuck at identity … and doesn’t move afterward

Pathological behavior suggesting sharp local minima!

Introduction: basic example

Why transformers fail?

And no, tuning/changing the optimizer doesn't help to solve this!



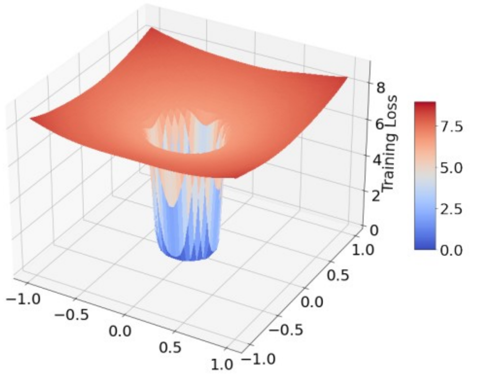
(a) SGD and AdamW with wd $= 1e{-}3$

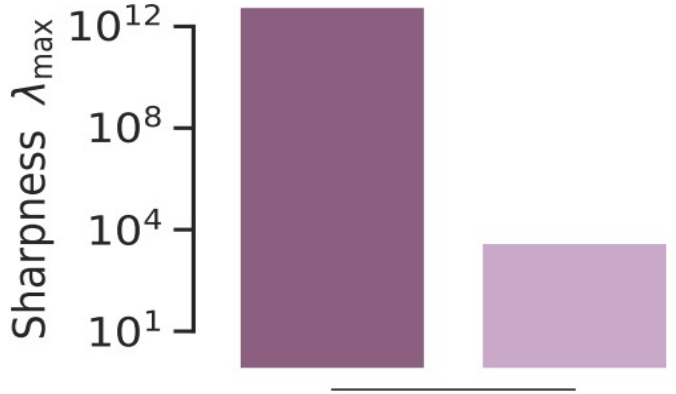
(b) AdamW with wd $\in \{1e{-}5, 1e{-}4\}$.

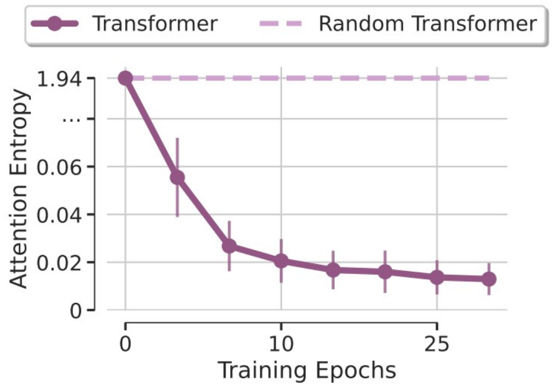
Introduction: basic example

Why transformers fail?

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







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

HUAWEI

Introduction: basic example

How to fix this?

1. **Reparametrization** (Zhen et al. 2023)

   - make attention matrix "more uniform" to avoid entropy collapse

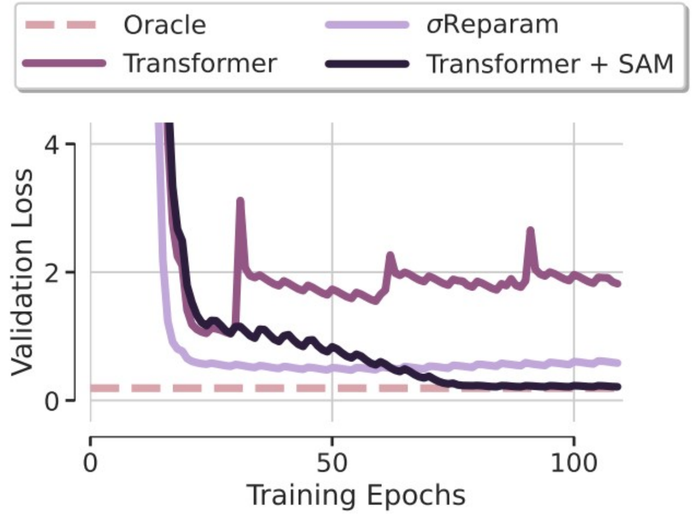
$$\widehat{\mathbf{W}} = \frac{\gamma}{\|\mathbf{W}\|_2}\mathbf{W}$$

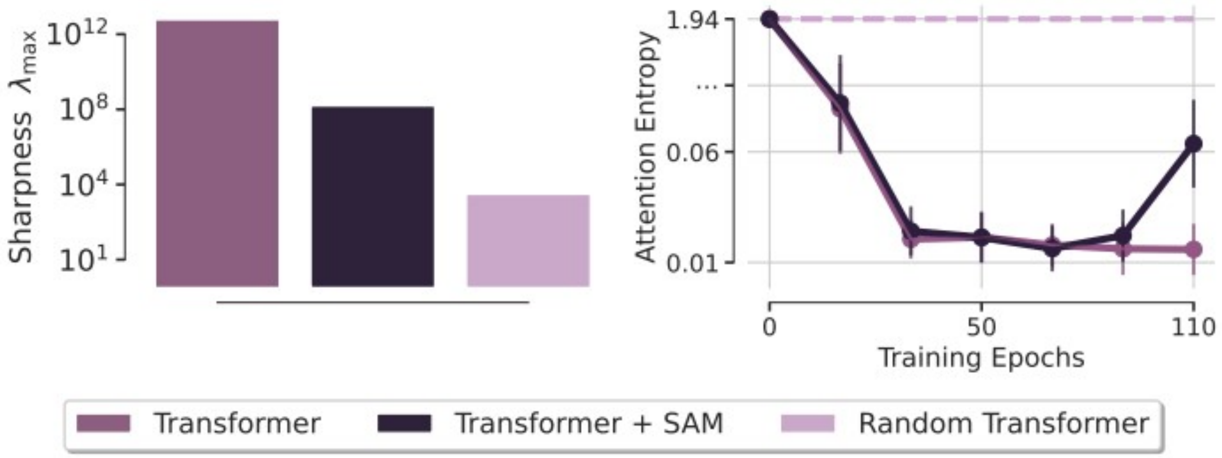
2. **Sharpness-aware minimization** (Foret et al. 2021, Chen et al. 2022)

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

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

Introduction: basic example





Observations:

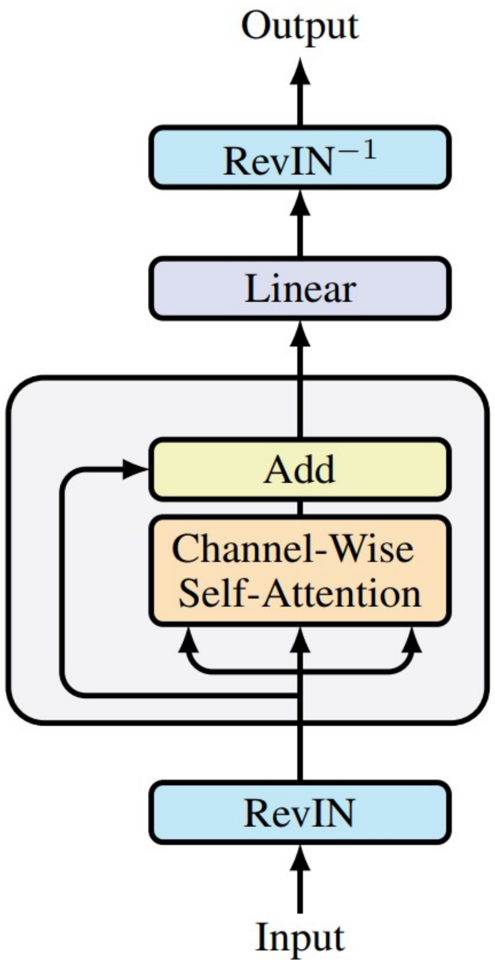
1. Reparametrization helps a bit!
2. Optimizing with **SAM = desired solution!**

Congrats you now know how to solve a linear regression problem with transformers!

Proposed model: SAMformer

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**



Experimental results: SAMformer

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: Informer (AAAI'21), FEDformer (ICML'22), Pyraformer (ICLR'22), Autoformer (NeurIPS'21), LogTrans (NeurIPS'19)

Experimental results: SAMformer

	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

   - much better than all transformer-based models

2. Sharpness-aware minimization **improves** TSMixer as well

Experimental results: SAMformer

SAMFormer is **smaller** and **more consistent** than TSMixer

- the **same model** for all datasets/horizons

- Avg Ratio = nbre params TSMixer / nbre params SAMFormer

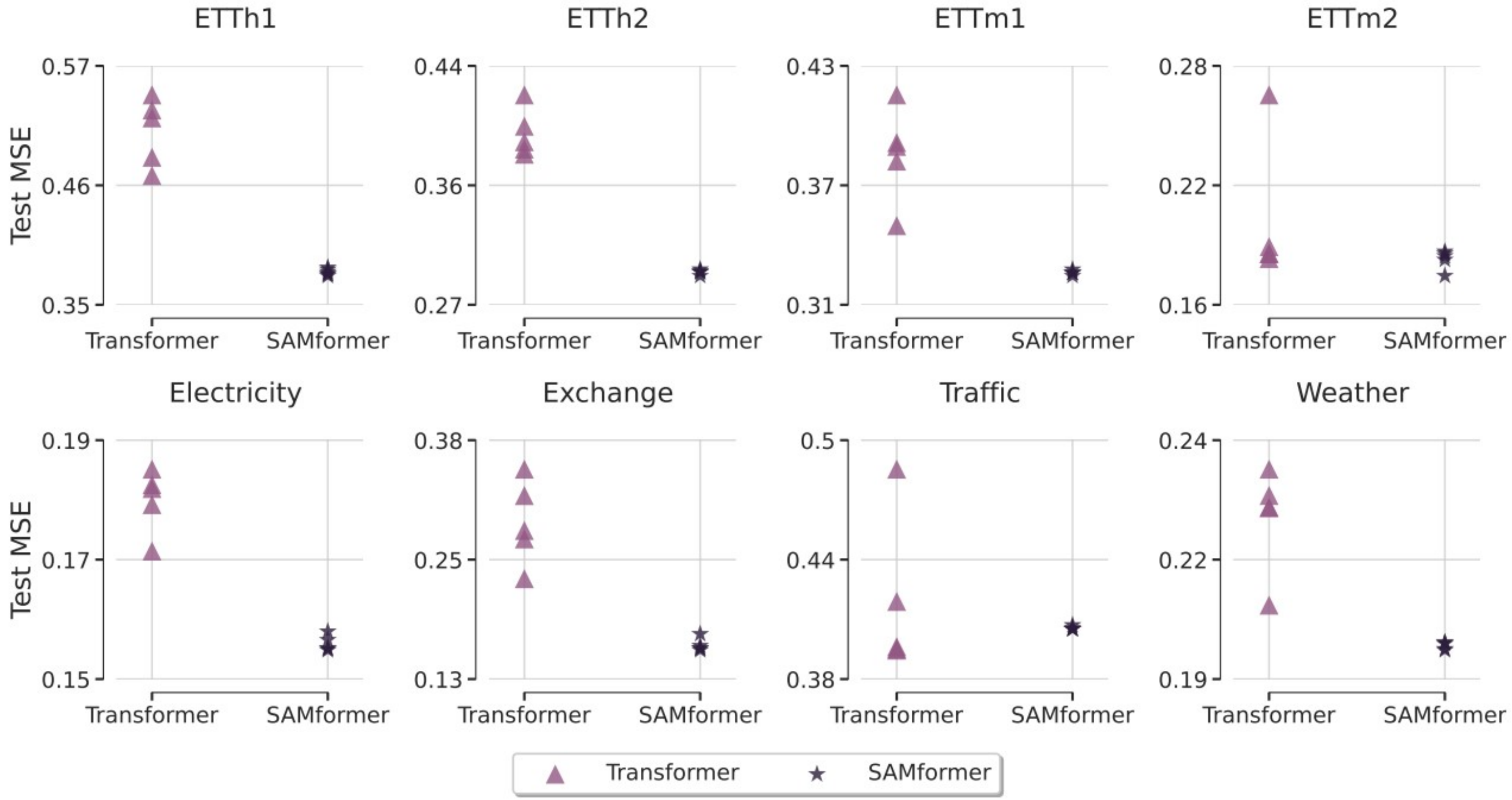
Dataset	$H = 96$		$H = 192$		$H = 336$		$H = 720$		Total
	SAMformer	TSMixer	SAMformer	TSMixer	SAMformer	TSMixer	SAMformer	TSMixer	
ETT	50272	124142	99520	173390	173392	247262	369904	444254	-
Exchange	50272	349344	99520	398592	173392	472464	369904	669456	-
Weather	50272	121908	99520	171156	173392	245028	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**

HUAWEI

Experimental results: SAMformer

SAMFormer is **robust to random initialization**

- very low variance for random seeds



HUAWEI

Experimental results: SAMformer

SAMFormer is **on par with MORAI 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)

		MOIRAI$_{Small}$	MOIRAI$_{Base}$	MOIRAI$_{Large}$	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	

Experimental results: SAMformer

Ablation study: why channel-wise attention?

- Candidate 1: SAMformer with **temporal** attention (as used in all other transformers)
- Overall Improvement : Improvement of SAMFormer over Temporal

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	$0.556_{\pm 0.009}$	$0.442_{\pm 0.006}$	$0.584_{\pm 0.027}$	$0.459_{\pm 0.004}$	$0.396_{\pm 0.012}$	$0.782_{\pm 0.041}$	-	$0.356_{\pm 0.000}$	

Experimental results: SAMformer

Ablation study: why channel-wise attention?

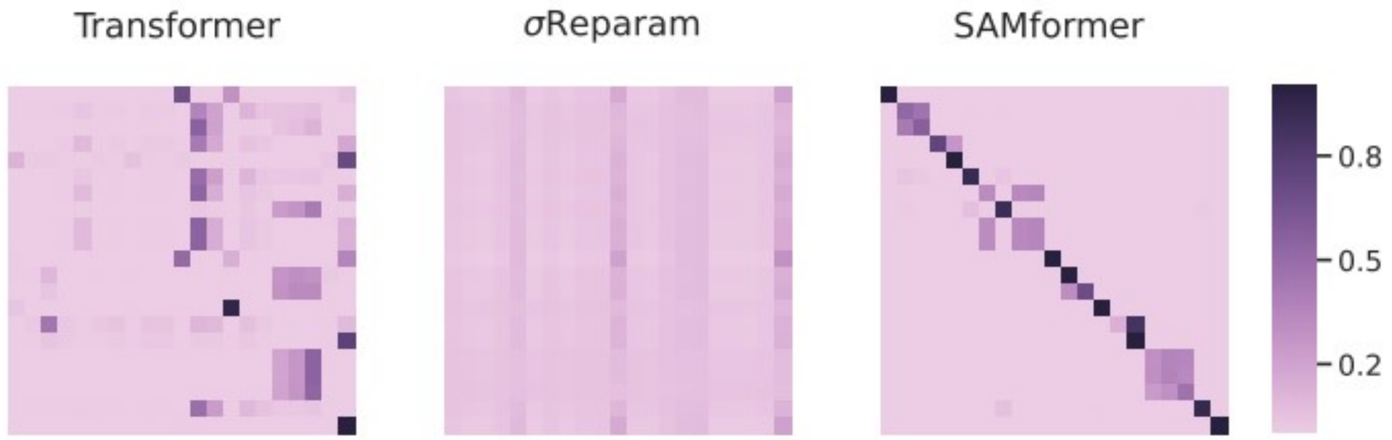
- Candidate 2: SAMformer with **identity weight matrix** attention
- Overall Improvement : Improvement of SAMFormer over Identity Attention

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	$0.496_{\pm 0.066}$	$0.438_{\pm 0.008}$	$0.444_{\pm 0.030}$	$0.406_{\pm 0.017}$	$0.322_{\pm 0.021}$	$0.788_{\pm 0.021}$	$0.325_{\pm 0.023}$	$0.347_{\pm 0.009}$	

SAM vs weight reparametrization

Final word on **weight matrix reparametrization**

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



Observations:

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

SAM vs weight reparametrization

Oversmoothing = rank collapse

- 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*

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

Roughly = **rank** of the attention matrix

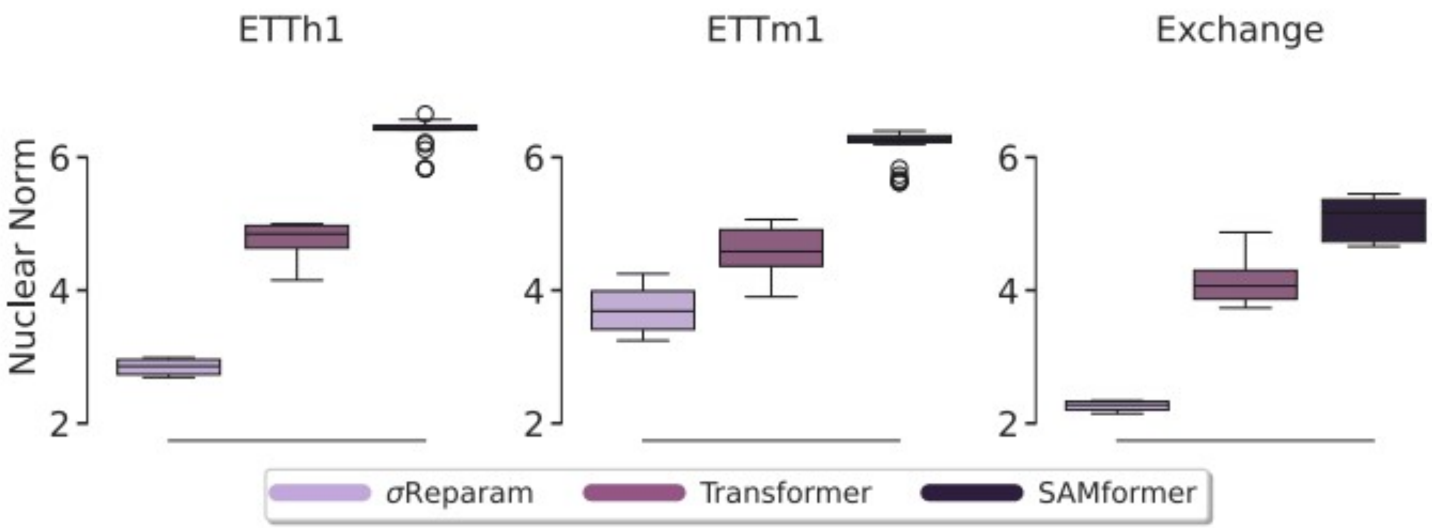
Minimized by reparametrization

- maximizing the entropy of the attention = rank collapse

- rank collapse = uninformative channel-wise attention

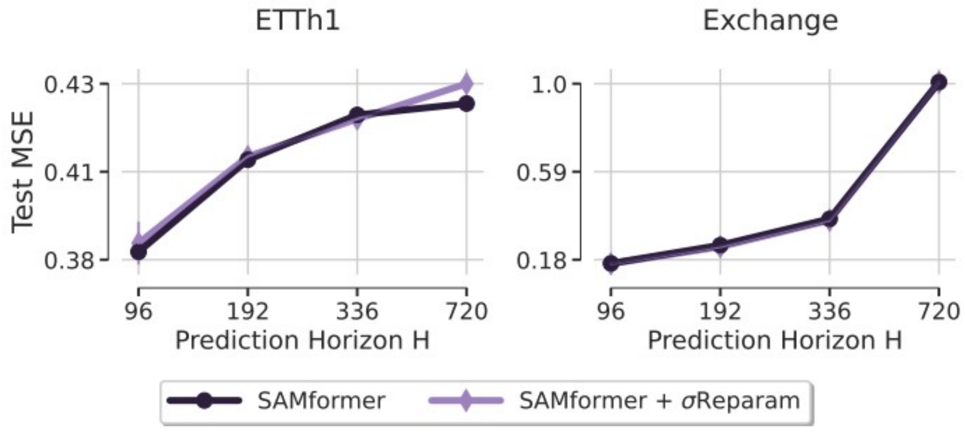
SAM vs weight reparametrization

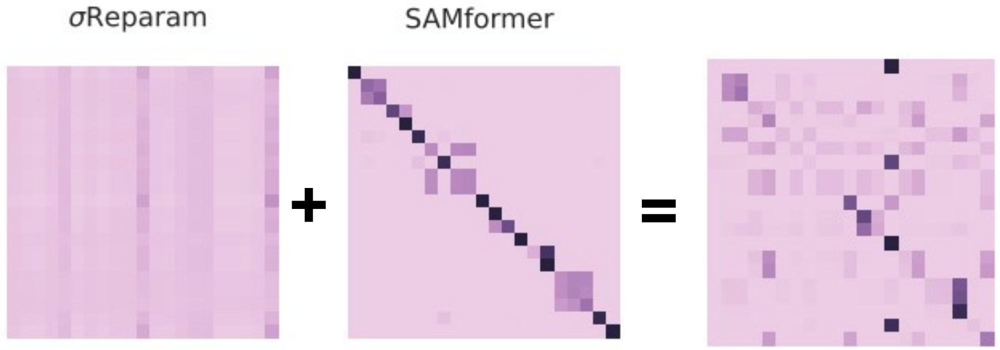
Oversmoothing = rank collapse



SAM + weight reparametrization

<u>A bit of smoothing + SAM</u> doesn't help much!





SAMformer: conclusions

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