Expand and Compress: Exploring ***Tuning Principles*** for Continual Spatio-Temporal Graph Forecasting —— ICLR 2025

Wei Chen Yuxuan Liang†
The Hong Kong University Of Science and Technology (Guangzhou)

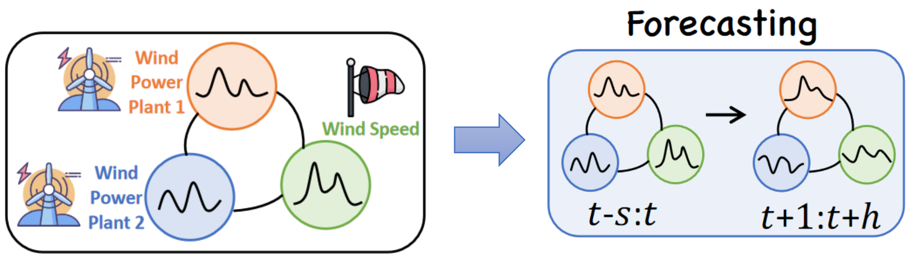
Wei Chen
Oct 21, 2024

Outline

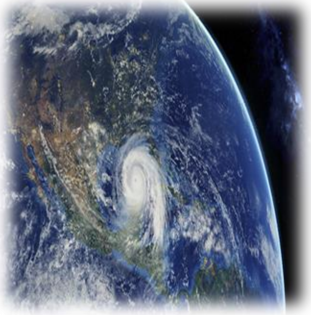
- 1. Introduction
- 2. Naive Solution Schemes
- 3. Our Solution
- 4. Tuning Principle I : Expand
- 5. Tuning Principle II : Compress
- 6. Experiment
- 7. Discussion
- 8. More Future work

Introduction

- **What is Spatio-Temporal Forecasting (STF) ?**
  - **It can be considered as a complex extended case of multivariate time series forecasting, where different variables have spatially dependent properties.**



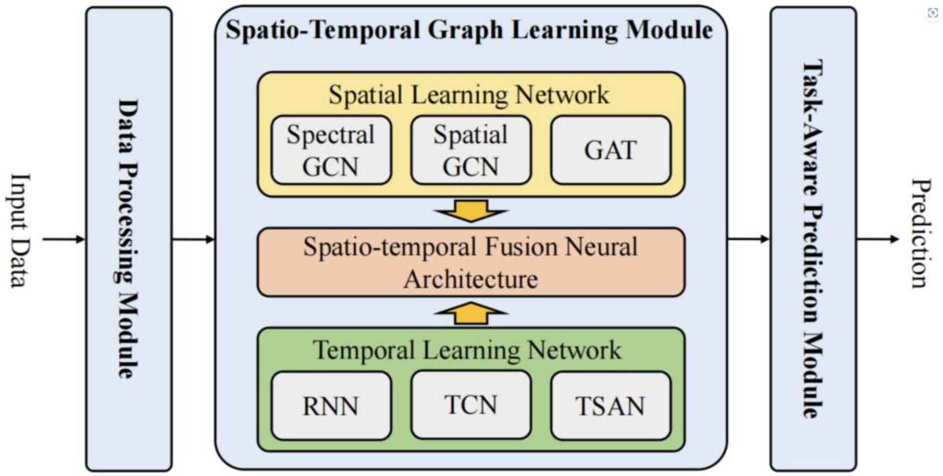
- **Application of Spatio-Temporal Forecasting**
  - **Traffic Flow Optimization**
  - **Air Quality Management**
  - **Win Energy Allocation**



Introduction

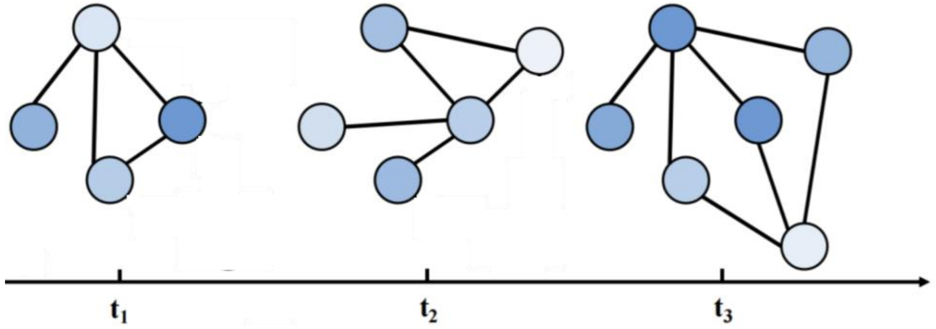
- How to modeling STF task ?
  - Spatial Modeling
    - Graph Operator
      - Spectral-Based / Spatial-Based
  - Temporal Modeling
    - Sequence Operator
      - RNN-Based / CNN (TCN) –Based / Attention-Based
  - Fusion them to focus on <u>spatio-temporal correlations</u>

STGNN


Input Data
Data Processing Module
Spatio-Temporal Graph Learning Module
Spatial Learning Network
Spectral GCN
Spatial GCN
GAT
Spatio-temporal Fusion Neural Architecture
Temporal Learning Network
RNN
TCN
TSAN
Task-Aware Prediction Module
Prediction

Introduction

- **Spatio-Temporal Forecasting in Real-World ?**
  - **Dynamic! & Streaming!**



- **Problem Define:**
  - Continual Spatio-temporal Graph Forecasting

$$f_{\theta^{(\tau)*}} = \operatorname*{argmin}_{\theta^{(\tau)}} \mathbb{E}_{D_\tau \sim \mathcal{P}^{(\tau)}} [\mathcal{L}(f_{\theta^{(\tau)}}(\mathcal{G}_\tau, X_\tau), Y_\tau)]$$

Naive solution schemes

- ***Pre-training***
  - suffer from distribution shifts， fails to adapt to new period data.
- ***Re-training***
  - neglects the informational gains from historical data, limited performance improvements
- ***Online-training*** 🚩
  - still inefficiency of retraining models over newly-arrived data
  - detrimental effects of catastrophic forgetting over long-term history

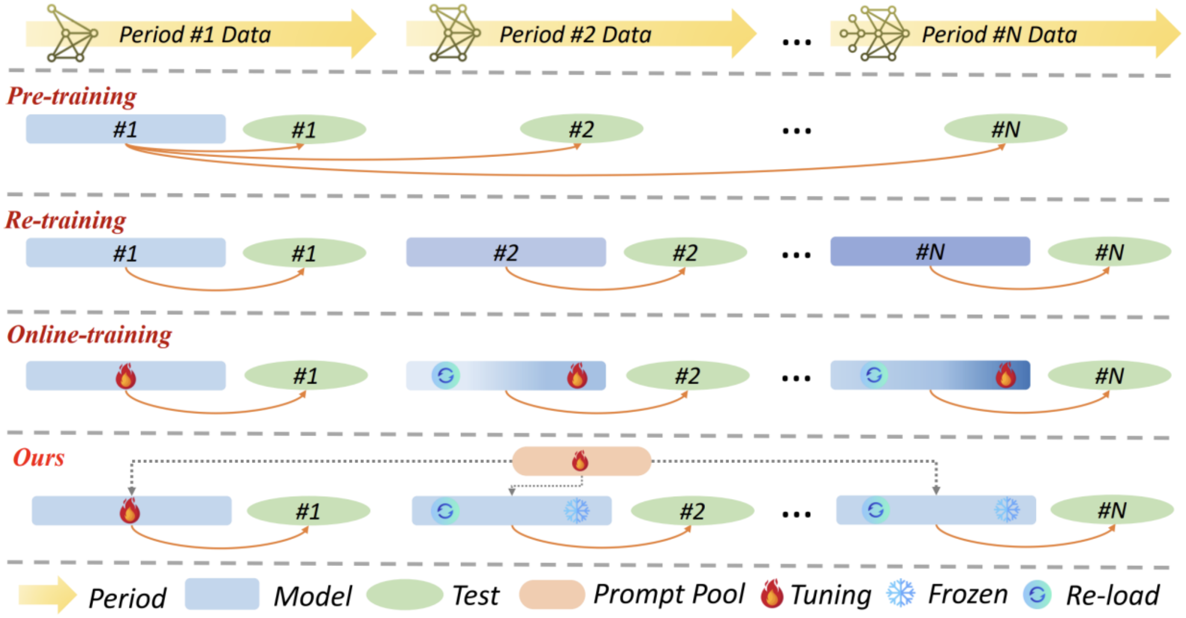


Figure 1: Comparison of classic schemes and EAC for continual spatio-temporal forecasting.

Our Solution

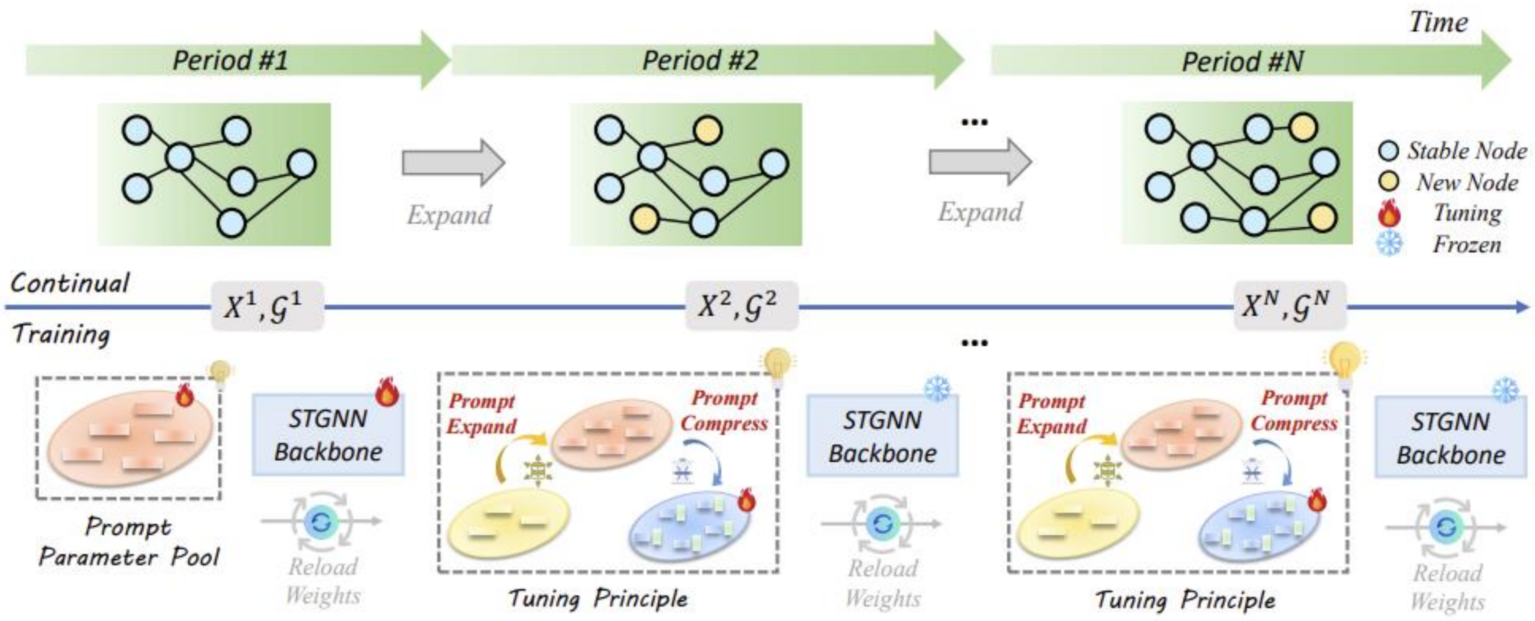
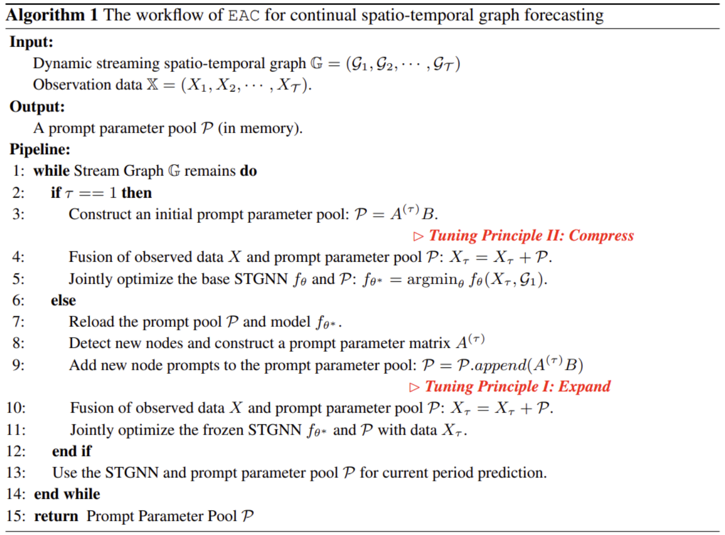


Figure 2: The overall architecture of our proposed EAC .



```
Algorithm 1 The workflow of EAC for continual spatio-temporal graph forecasting
Input:
    Dynamic streaming spatio-temporal graph $\mathbb{G} = (\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_T)$
    Observation data $\mathbb{X} = (X_1, X_2, \cdots, X_T)$.
Output:
    A prompt parameter pool $\mathcal{P}$ (in memory).
Pipeline:
while Stream Graph $\mathbb{G}$ remains do
    if $\tau == 1$ then
        Construct an initial prompt parameter pool: $\mathcal{P} = A^{(\tau)}B$.
                                                ▷ Tuning Principle II: Compress
        Fusion of observed data $X$ and prompt parameter pool $\mathcal{P}$: $X_\tau = X_\tau + \mathcal{P}$.
        Jointly optimize the base STGNN $f_\theta$ and $\mathcal{P}$: $f_{\theta^*} = \text{argmin}_\theta f_\theta(X_\tau, \mathcal{G}_1)$.
    else
        Reload the prompt pool $\mathcal{P}$ and model $f_{\theta^*}$.
        Detect new nodes and construct a prompt parameter matrix $A^{(\tau)}$
        Add new node prompts to the prompt parameter pool: $\mathcal{P} = \mathcal{P}.append(A^{(\tau)}B)$
                                                ▷ Tuning Principle I: Expand
        Fusion of observed data $X$ and prompt parameter pool $\mathcal{P}$: $X_\tau = X_\tau + \mathcal{P}$.
        Jointly optimize the frozen STGNN $f_{\theta^*}$ and $\mathcal{P}$ with data $X_\tau$.
    end if
    Use the STGNN and prompt parameter pool $\mathcal{P}$ for current period prediction.
end while
return Prompt Parameter Pool $\mathcal{P}$
```

Tuning Principle I : Expand

- **Motivation -> Solution**

Inefficiency / Catastrophic Forgetting → *A straightforward solution is to* ***isolate parameters, freeze the old model, and dynamically adjust the network structure to incorporate adaptable learning parameters****.*

- **Insight**

Trend: *node-specific trainable parameters as spatial identifiers can achieve higher performance.*

⬇

why they are useful? when they are applicable? and in what contexts they are most suitable?

⬇

Intuition : Spatio-temporal data generally exhibit two characteristics: ***correlation*** and ***heterogeneity*** ⭐

$$\mathbf{Z} = \mathbf{U} g_\theta(\mathbf{\Lambda})\mathbf{U}^\top \mathbf{X}$$

$$\mathbf{Z} = \sigma\left(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}\right)$$

✅ ↖ ↗ ❌

message-passing mechanism of STGNNs

Tuning Principle I : Expand

- ## Empirical Observation

To quantitatively analyze heterogeneity, we consider the ***dispersion of node feature vectors in the feature space***

$$D(X) = \frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{d} (X_{ik} - X_{jk})^2$$

Two observation:

① Within the same period, the dispersion of the node feature space continuously expand

② Across different periods, the dispersion of the feature space in the current period expand further compared to the previous

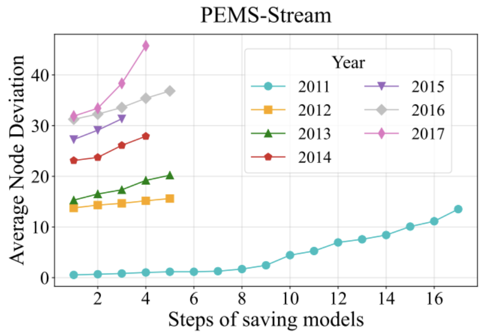
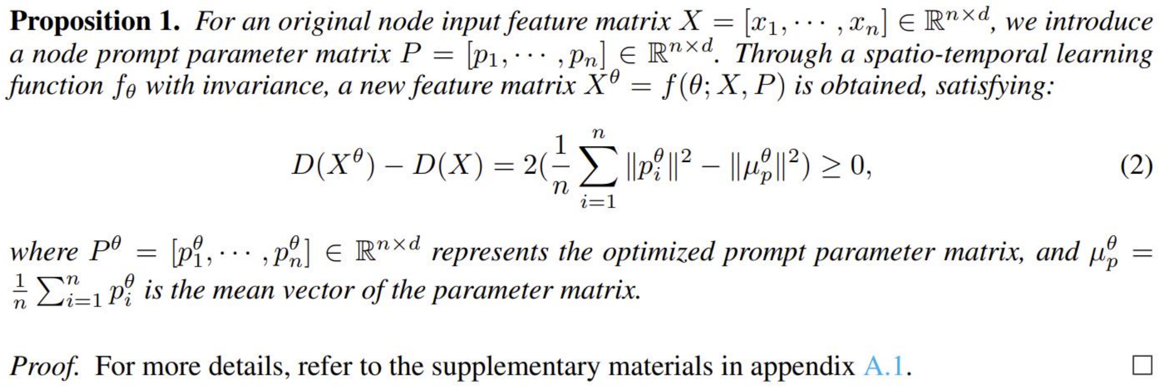


Figure 3: Heterogeneity measurement.

- ## Theoretical Analysis



Proposition 1. *For an original node input feature matrix $X = [x_1, \cdots, x_n] \in \mathbb{R}^{n \times d}$, we introduce a node prompt parameter matrix $P = [p_1, \cdots, p_n] \in \mathbb{R}^{n \times d}$. Through a spatio-temporal learning function f_θ with invariance, a new feature matrix $X^\theta = f(\theta; X, P)$ is obtained, satisfying:*

$$D(X^\theta) - D(X) = 2(\frac{1}{n}\sum_{i=1}^{n} \|p_i^\theta\|^2 - \|\mu_p^\theta\|^2) \geq 0, \tag{2}$$

where $P^\theta = [p_1^\theta, \cdots, p_n^\theta] \in \mathbb{R}^{n \times d}$ represents the optimized prompt parameter matrix, and $\mu_p^\theta = \frac{1}{n}\sum_{i=1}^{n} p_i^\theta$ is the mean vector of the parameter matrix.

Proof. For more details, refer to the supplementary materials in appendix A.1. □

***Tuning Principle I:** Prompt Parameter Pool Can Continuously Adapt to Heterogeneity Property.*

Tuning Principle II : Compress

- **Motivation -> Solution**

Parameter Inflation → *An intuitive solution is to similarly apply low-rank matrix approximations to the prompt parameter pool, thereby reducing the number of learnable parameters while maintaining performance.*

- **Insight**

Background: numerous well-established studies that enhance the efficiency of spatio-temporal prediction and imputation tasks using compressed sensing and matrix / tensor decomposition

⬇

These study typically focus solely on the raw ST-data, how about prompt parameter pool ?

⬇

Intuition : If the prompt parameter pool also exhibits ***similar redundancy*** characteristics and ***remains in the continuous setting***, then we are good.

Tuning Principle II : Compress

- ## Empirical Observation

To explore redundancy, we conduct a spectral analysis of the prompt parameter pool.

$$\mathbf{A}_t = \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^\top \quad \Sigma_t = \mathrm{diag}(\sigma_{t,1}, \sigma_{t,2}, \ldots, \sigma_{t,r_t})$$

$$C_t^{\mathrm{norm}}(k) = \frac{\sum_{i=1}^{k} \sigma_{t,i}}{\sum_{i=1}^{r_t} \sigma_{t,i}}, \quad k = 1, 2, \ldots, r_t, \quad t = 1, 2, \ldots, T$$

Two observation:

① All years exhibit a clear long-tail spectral distribution

② The overall processes for all years maintain a high concentration of information (> 0.75

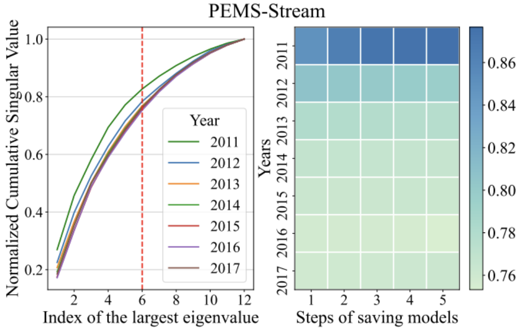


Figure 4: Low-rank measurement.

- ## Theoretical Analysis

Proposition 2. *Given the node prompt parameter matrix $P \in \mathbb{R}^{n \times d}$, there will always be two matrices $A \in \mathbb{R}^{n \times k}$ and $B \in \mathbb{R}^{k \times d}$ such that P can be approximated as AB when the nodes n grow large, and satisfy the following probability inequality:*

$$\Pr\left(\|P - AB\|_F \le \epsilon \|P\|_F\right) \ge 1 - o(1) \ \textit{ and } \ k = \mathcal{O}\left(\log(\min(n, d))\right)$$

where $o(1)$ represents a term that becomes negligible even as n grows large.

Proof. For more details, refer to the supplementary materials in appendix A.2. □

***Tuning Principle II:** Prompt Parameter Pool Can Continuously Satisfy the Low-rank Property.*

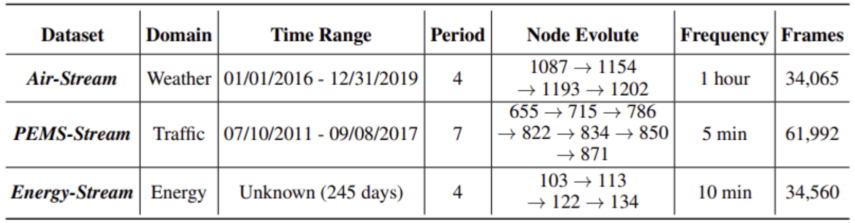
Experiment

- **RQ1 (Effectiveness) :** Can EAC outperform previous methods in accuracy across various tasks?
- **RQ2 (Universality) :** Can EAC have a consistent improvement on various types of STGNNs?
- **RQ3 (Efficiency) :** How efficient is EAC compared to different methods during the training phase?
- **RQ4 (Lightweight) :** How many parameters does EAC require tuning compared to baselines?
- **RQ5 (Simplicity) :** How does EAC compare to other common prompt-adaptive learning method?

Experiment

- **Dataset and Evaluation Protocol.**

Table 5: Summary of datasets used for continual spatio-temporal datasets.



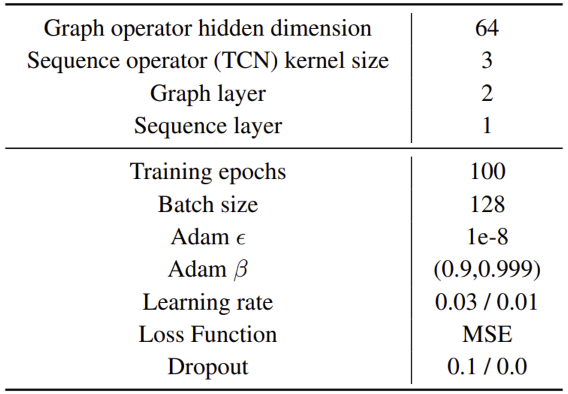
Dataset	Domain	Time Range	Period	Node Evolute	Frequency	Frames
Air-Stream	Weather	01/01/2016 - 12/31/2019	4	1087 → 1154 → 1193 → 1202	1 hour	34,065
PEMS-Stream	Traffic	07/10/2011 - 09/08/2017	7	655 → 715 → 786 → 822 → 834 → 850 → 871	5 min	61,992
Energy-Stream	Energy	Unknown (245 days)	4	103 → 113 → 122 → 134	10 min	34,560

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \quad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \quad \text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$$

- **Baseline and Parameter Setting.**

- Pre-train-ST
- Re-train-ST
- Online-ST
  - Online-ST-AN
  - Online-ST-NN: TFMoE
  - Online-ST-MN: TrafficStream / PECPM / STKEC

Table 6: Hyperparameters setting.



Graph operator hidden dimension	64
Sequence operator (TCN) kernel size	3
Graph layer	2
Sequence layer	1
Training epochs	100
Batch size	128
Adam ϵ	1e-8
Adam β	(0.9,0.999)
Learning rate	0.03 / 0.01
Loss Function	MSE
Dropout	0.1 / 0.0

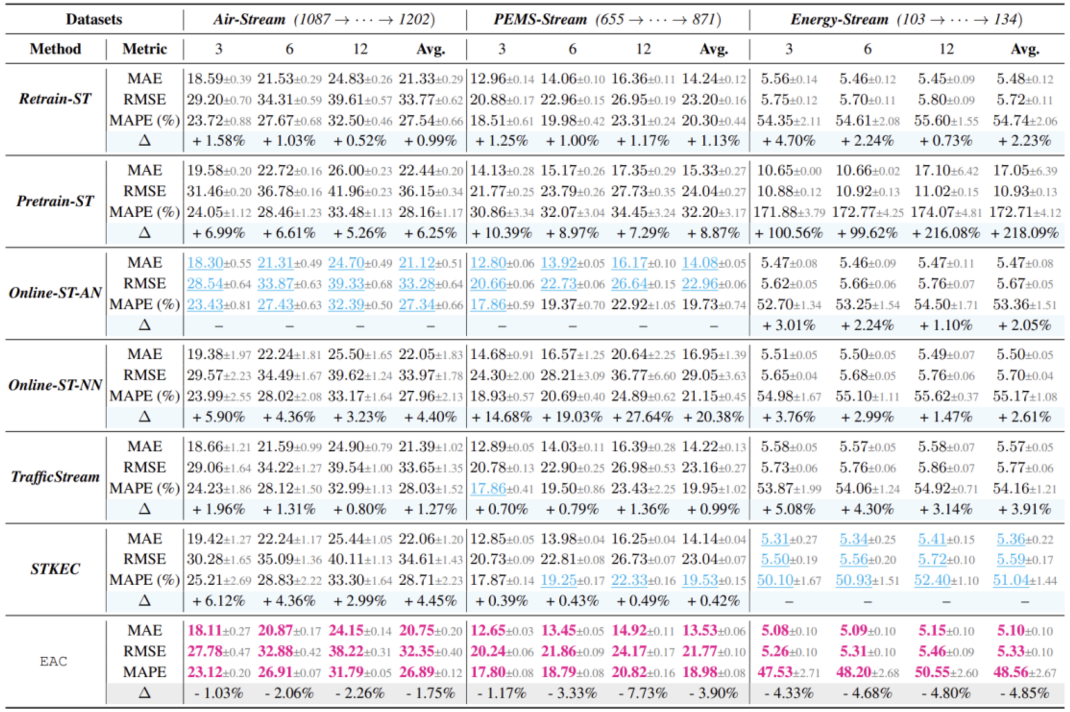
Same STGNN Backbone

The only hyperparameter: k

Experiment

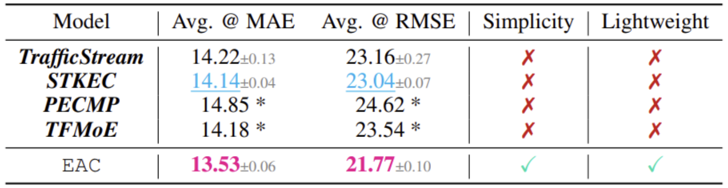
- **Overall Performance (RQ1)**
  - ❶ Pretrain-ST methods generally yield the poorest results, especially on smaller datasets
  - ❷ Retrain-ST methods also exhibit unsatisfactory results, as they not effectively utilizing historical information
  - ❸ Online-ST-NN methods perform poorly, as they finetune the pretrained model using only new node data. This leads to catastrophic forgetting
  - ❹ Online-ST-MN methods strike a balance between performance and efficiency, showing some improvements
  - ❺ Online-ST-AN methods typically achieve suboptimal results, approximating the performance boundary. But this is not efficient enough
  - ❻ Our EAC consistently improves all metrics across all types of datasets.

Table 1: Comparison of the overall performance of the classical scheme and EAC .



Datasets		*Air-Stream* (*1087* → ··· → *1202*)				*PEMS-Stream* (*655* → ··· → *871*)				*Energy-Stream* (*103* → ··· → *134*)			
Method	**Metric**	3	6	12	**Avg.**	3	6	12	**Avg.**	3	6	12	**Avg.**
Retrain-ST	MAE	18.59±0.39	21.53±0.29	24.83±0.26	21.33±0.29	12.96±0.14	14.06±0.10	16.36±0.11	14.24±0.12	5.56±0.14	5.46±0.12	5.45±0.09	5.48±0.12
	RMSE	29.20±0.70	34.31±0.59	39.61±0.57	33.77±0.62	20.88±0.17	22.96±0.15	26.95±0.19	23.20±0.16	5.75±0.12	5.70±0.11	5.80±0.09	5.72±0.11
	MAPE (%)	23.72±0.88	27.67±0.68	32.50±0.46	27.54±0.66	18.51±0.61	19.98±0.42	23.31±0.24	20.30±0.44	54.35±2.11	54.61±2.08	55.60±1.55	54.74±2.06
	Δ	+ 1.58%	+ 1.03%	+ 0.52%	+ 0.99%	+ 1.25%	+ 1.00%	+ 1.17%	+ 1.13%	+ 4.70%	+ 2.24%	+ 0.73%	+ 2.23%
Pretrain-ST	MAE	19.58±0.20	22.72±0.16	26.00±0.23	22.44±0.20	14.13±0.28	15.17±0.26	17.35±0.29	15.33±0.27	10.65±0.00	10.66±0.02	17.10±6.42	17.05±6.39
	RMSE	31.46±0.20	36.78±0.16	41.96±0.23	36.15±0.34	21.77±0.25	23.79±0.26	27.73±0.35	24.04±0.27	10.88±0.12	10.92±0.13	11.02±0.15	10.93±0.13
	MAPE (%)	24.05±1.12	28.46±1.23	33.48±1.13	28.16±1.17	30.86±3.34	32.07±3.04	34.45±3.24	32.20±3.17	171.88±3.79	172.77±4.25	174.07±4.81	172.71±4.12
	Δ	+ 6.99%	+ 6.61%	+ 5.26%	+ 6.25%	+ 10.39%	+ 8.97%	+ 7.29%	+ 8.87%	+ 100.56%	+ 99.62%	+ 216.08%	+ 218.09%
Online-ST-AN	MAE	18.30±0.55	21.31±0.49	24.70±0.49	21.12±0.51	12.80±0.06	13.92±0.05	16.17±0.10	14.08±0.05	5.47±0.08	5.46±0.09	5.47±0.11	5.47±0.08
	RMSE	28.54±0.64	33.87±0.63	39.33±0.68	33.28±0.64	20.66±0.06	22.73±0.06	26.64±0.15	22.96±0.06	5.62±0.05	5.66±0.06	5.76±0.07	5.67±0.05
	MAPE (%)	23.43±0.81	27.43±0.63	32.39±0.50	27.34±0.66	17.86±0.59	19.37±0.70	22.92±1.05	19.73±0.74	52.70±1.34	53.25±1.54	54.50±1.71	53.36±1.51
	Δ	–	–	–	–	–	–	–	–	+ 3.01%	+ 2.24%	+ 1.10%	+ 2.05%
Online-ST-NN	MAE	19.38±1.97	22.24±1.81	25.50±1.65	22.05±1.83	14.68±0.91	16.57±1.25	20.64±2.25	16.95±1.39	5.51±0.05	5.50±0.05	5.49±0.07	5.50±0.05
	RMSE	29.57±2.23	34.49±1.67	39.62±1.24	33.97±1.78	24.30±2.00	28.21±3.09	36.77±6.60	29.05±3.63	5.65±0.04	5.68±0.05	5.76±0.06	5.70±0.04
	MAPE (%)	23.99±2.55	28.02±2.08	33.17±1.64	27.96±2.13	18.93±0.57	20.69±0.40	24.89±0.62	21.15±0.45	54.98±1.67	55.10±1.11	55.62±0.37	55.17±1.08
	Δ	+ 5.90%	+ 4.36%	+ 3.23%	+ 4.40%	+ 14.68%	+ 19.03%	+ 27.64%	+ 20.38%	+ 3.76%	+ 2.99%	+ 1.47%	+ 2.61%
TrafficStream	MAE	18.66±1.21	21.59±0.99	24.90±0.79	21.39±1.02	12.89±0.05	14.03±0.11	16.39±0.28	14.22±0.13	5.58±0.05	5.57±0.05	5.58±0.07	5.57±0.05
	RMSE	29.06±1.64	34.22±1.27	39.54±1.00	33.65±1.35	20.78±0.13	22.90±0.25	26.98±0.53	23.16±0.27	5.73±0.06	5.76±0.06	5.86±0.07	5.77±0.06
	MAPE (%)	24.23±1.86	28.12±1.50	32.99±1.13	28.03±1.52	17.86±0.41	19.50±0.86	23.43±2.25	19.95±1.02	53.87±1.99	54.06±1.24	54.92±0.71	54.16±1.21
	Δ	+ 1.96%	+ 1.31%	+ 0.80%	+ 1.27%	+ 0.70%	+ 0.79%	+ 1.36%	+ 0.99%	+ 5.08%	+ 4.30%	+ 3.14%	+ 3.91%
STKEC	MAE	19.42±1.27	22.24±1.17	25.44±1.05	22.06±1.20	12.85±0.05	13.98±0.04	16.25±0.04	14.14±0.04	5.31±0.27	5.34±0.25	5.41±0.15	5.36±0.22
	RMSE	30.28±1.65	35.09±1.36	40.11±1.13	34.61±1.43	20.73±0.09	22.81±0.08	26.73±0.07	23.04±0.07	5.50±0.19	5.56±0.20	5.72±0.10	5.59±0.17
	MAPE (%)	25.21±2.69	28.83±2.22	33.30±1.64	28.71±2.23	17.87±0.14	19.25±0.17	22.33±0.16	19.53±0.15	50.10±1.67	50.93±1.51	52.40±1.10	51.04±1.44
	Δ	+ 6.12%	+ 4.36%	+ 2.99%	+ 4.45%	+ 0.39%	+ 0.43%	+ 0.49%	+ 0.42%	–	–	–	–
EAC	MAE	**18.11**±0.27	**20.87**±0.17	**24.15**±0.14	**20.75**±0.20	**12.65**±0.03	**13.45**±0.05	**14.92**±0.11	**13.53**±0.06	**5.08**±0.10	**5.09**±0.10	**5.15**±0.10	**5.10**±0.10
	RMSE	**27.78**±0.47	**32.88**±0.42	**38.22**±0.31	**32.35**±0.40	**20.24**±0.06	**21.86**±0.09	**24.17**±0.17	**21.77**±0.10	**5.26**±0.10	**5.31**±0.10	**5.46**±0.09	**5.33**±0.10
	MAPE	**23.12**±0.20	**26.91**±0.07	**31.79**±0.05	**26.89**±0.12	**17.80**±0.08	**18.79**±0.08	**20.82**±0.16	**18.98**±0.08	**47.53**±2.71	**48.20**±2.68	**50.55**±2.60	**48.56**±2.67
	Δ	- 1.03%	- 2.06%	- 2.26%	- 1.75%	- 1.17%	- 3.33%	- 7.73%	- 3.90%	- 4.33%	- 4.68%	- 4.80%	- 4.85%

Table 2: Performance comparison of the improved method with EAC on *PEMS-Stream* benchmark.



Model	Avg. @ MAE	Avg. @ RMSE	Simplicity	Lightweight
TrafficStream	14.22±0.13	23.16±0.27	✗	✗
STKEC	14.14±0.04	23.04±0.07	✗	✗
PECMP	14.85 *	24.62 *	✗	✗
TFMoE	14.18 *	23.54 *	✗	✗
EAC	**13.53**±0.06	**21.77**±0.10	✓	✓

Experiment

- **Few-Shot Performance (RQ1)**
  - ❶ (Robustness) All methods exhibit a decline in performance compared to the complete data scenario, with Online-ST-NN demonstrating significant sensitivity due to its inherent catastrophic forgetting issue.
  - ❷ (Adaptability) The performance of all methods consistently declines as the periods extend, yet our EAC method demonstrates a relatively mild decline

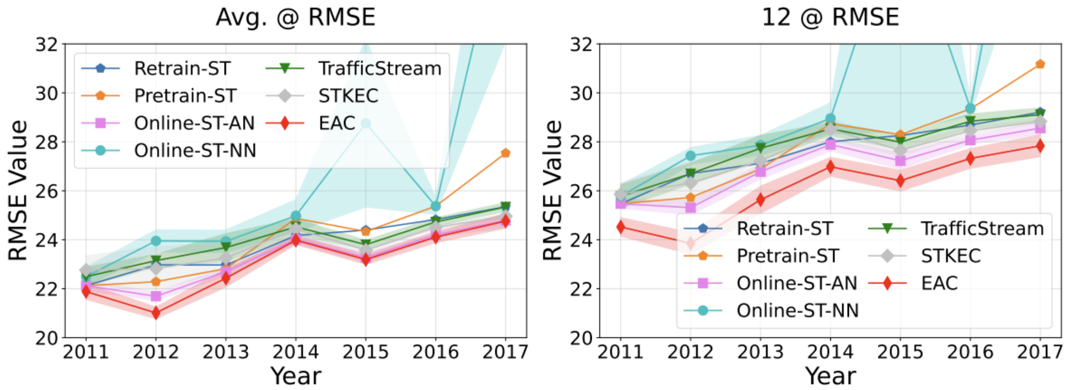
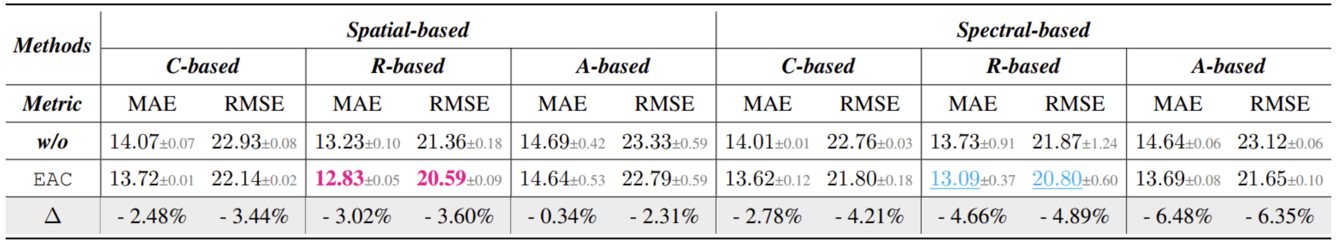


Figure 5: Few-Shot Scenario Forecasting in *PEMS-Stream* benchmark.

Experiment

- **Universality Study (RQ2)**
  - ❶ Our EAC consistently demonstrates performance improvements across different combinations, highlighting its universality for various architectures.
  - ❷ Compared to spatial domain-based graph convolution operators, our EAC shows a more pronounced enhancement for spectral domain-based methods.
  - ❸ The advantages of recurrent-based sequence modeling operators are particularly evident, achieving the best results, while attention-based methods perform the worst. Our EAC provides certain gains in all cases.

Table 3: Effect of EAC on the average performance of different STGNN component on the ***PEMS-Stream***. C-based: Convolution-based, R-based: Recurrent-based, A-based: Attention-based.



Methods	*Spatial-based*						*Spectral-based*					
	C-based		*R-based*		*A-based*		*C-based*		*R-based*		*A-based*	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o	14.07±0.07	22.93±0.08	13.23±0.10	21.36±0.18	14.69±0.42	23.33±0.59	14.01±0.01	22.76±0.03	13.73±0.91	21.87±1.24	14.64±0.06	23.12±0.06
EAC	13.72±0.01	22.14±0.02	**12.83**±0.05	**20.59**±0.09	14.64±0.53	22.79±0.59	13.62±0.12	21.80±0.18	13.09±0.37	20.80±0.60	13.69±0.08	21.65±0.10
Δ	- 2.48%	- 3.44%	- 3.02%	- 3.60%	- 0.34%	- 2.31%	- 2.78%	- 4.21%	- 4.66%	- 4.89%	- 6.48%	- 6.35%

Experiment

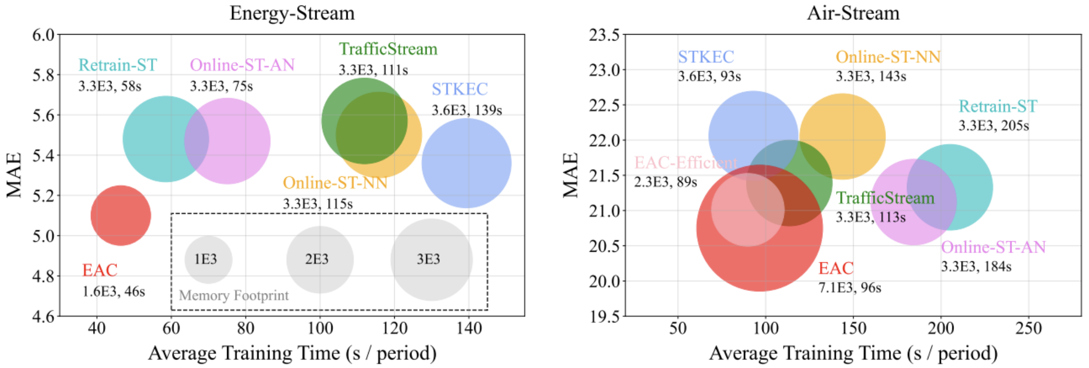


Figure 6: Efficiency & Lightwight & Performance Study.

- **Efficiency & Lightweight Study (RQ3 & RQ4)**
  - Overall Analysis.
    - ❶ On datasets with a smaller number of nodes, our EAC consistently outperforms the others, achieving superior performance with only half the number of tuning parameters. Furthermore, the average training time per period accelerates by a factor of 1.26 to 3.02.
    - ❷ On datasets with a larger number of nodes, although the EAC exhibits a slightly higher number of tuning parameters, the freezing of the backbone model results in faster training speeds while still achieving superior performance.
    - ❸ According to the tuning principle of compression, we set k = 2 to replace 6, resulting in the EAC - Efficient version, which maintains relative performance superiority on larger datasets using only ~ 63% parameters compared to others.

Experiment

- **Efficiency & Lightweight Study (RQ3 & RQ4)**
  - Hyper-parameter Analysis.

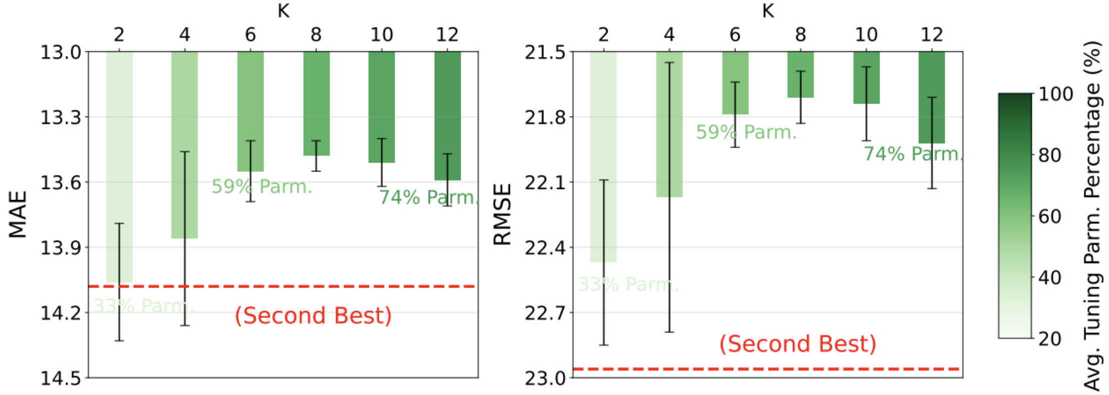


Figure 7: Hyper-parameter study in *PEMS-Stream* benchmark.

- ❶ From right to left, as the value of k decreases, indicating a reduction in tuning parameters, the overall performance of the model deteriorates significantly, accompanied by increased volatility (i.e., higher standard deviation). The effective representational information of the approximate prompt parameter pool is constrained with decreasing k.

- ❷ Although performance continues to improve with increasing k, the gains are minimal. Our method achieves satisfactory performance using only approximately 59% of the tuning parameters, effectively balancing performance and parameter efficiency.

Experiment

- **Simplicity Study (RQ5)**

  - ❶ Simply applying LoRA layers without considering the specific spatio-temporal context of streaming parameters may not be highly effective.

  - ❷ Our method enjoy shorter training times compared to LoRA-based approaches, further validating the superiority of our proposed expansion and compression tuning principle.

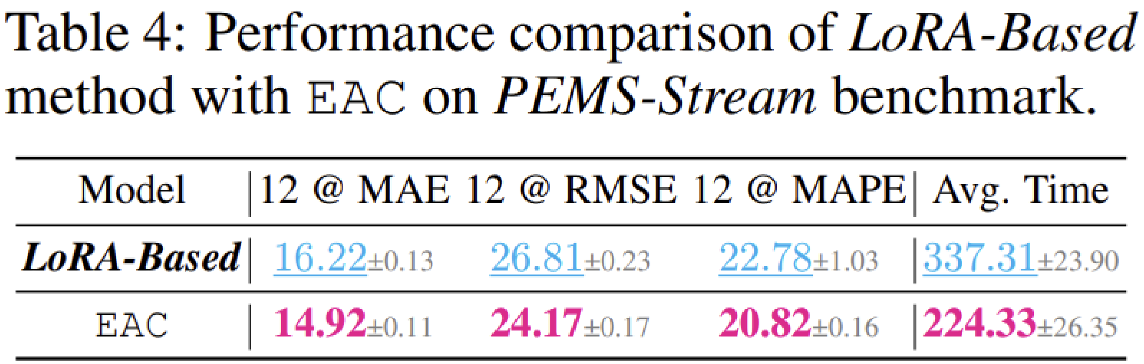


Table 4: Performance comparison of *LoRA-Based* method with EAC on *PEMS-Stream* benchmark.

Model	12 @ MAE	12 @ RMSE	12 @ MAPE	Avg. Time
LoRA-Based	16.22±0.13	26.81±0.23	22.78±1.03	337.31±23.90
EAC	**14.92**±0.11	**24.17**±0.17	**20.82**±0.16	**224.33**±26.35

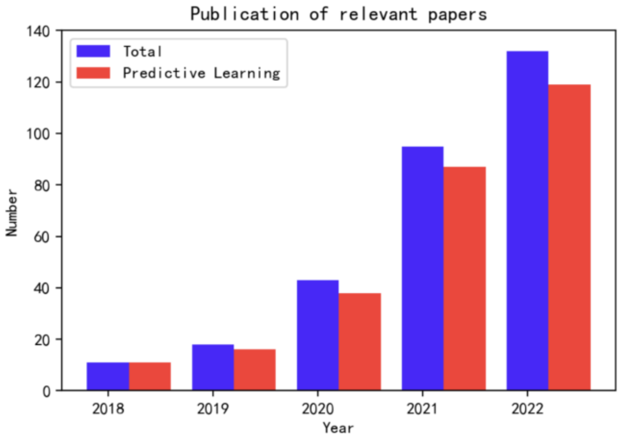
Discussion

- **Limitation**
  - ❶ All current baselines and datasets primarily focus on scenarios involving the continuous expansion of spatio-temporal graphs, with little consideration given to their reduction. -> Reasonable but not comprehensive, we can still handle it.
  - ❷ All current baselines and datasets span a maximum of seven years. -> Research extending beyond this time span remains worthy of exploration, but retraining seems more appropriate in some context.
  - ❸ Due to the node-level design of our prompt parameter pool, the issue of parameter bloat is inevitable. -> Other technical, such as parameter sparsification and pruning, also merit investigation. But we provide a compress principle.

- **Future Work**
  - ❶ One lesson learned from our experiments is that the initially pre-trained spatio-temporal graph model is crucial for the subsequent continuous fine-tuning process. -> **Large Foundation ST-Model** ⭐

Thank you!
Q & A

Reflection



- *Graph Community:*

  ChebNet (2016) → GCN(2017) → AGCN(2018)

  → GAT(2018) → GIN(2019)

- *NLP Community:*

  LSTM (1997) → TCN (2016) → Attention (2015)

  → Transformer (2017) → PLM (2020) → LLM (2022)

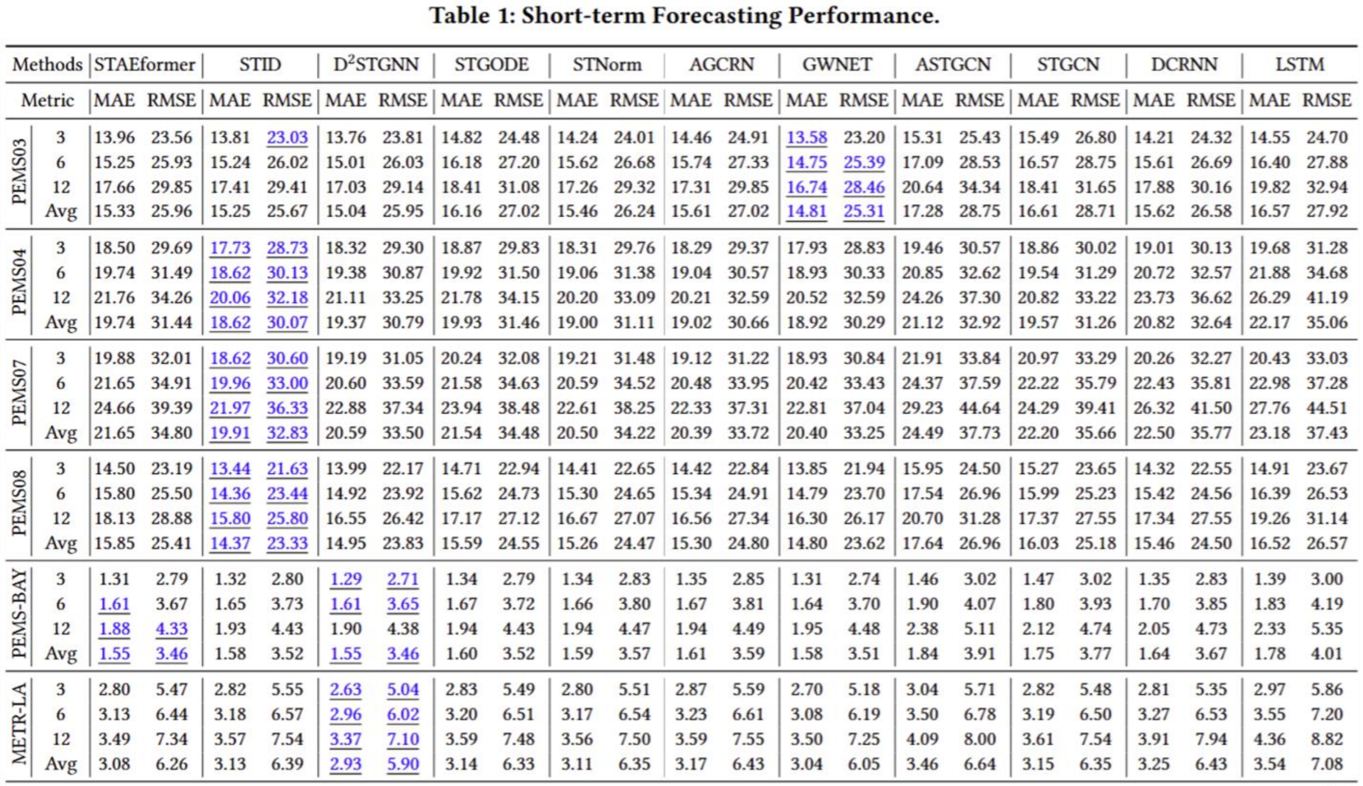


Table 1: Short-term Forecasting Performance.

Methods		STAEformer		STID		D²STGNN		STGODE		STNorm		AGCRN		GWNET		ASTGCN		STGCN		DCRNN		LSTM	
Metric		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PEMS03	3	13.96	23.56	13.81	23.03	13.76	23.81	14.82	24.48	14.24	24.01	14.46	24.91	13.58	23.20	15.31	25.43	15.49	26.80	14.21	24.32	14.55	24.70
	6	15.25	25.93	15.24	26.02	15.01	26.03	16.18	27.20	15.62	26.68	15.74	27.33	14.75	25.39	17.09	28.53	16.57	28.75	15.61	26.69	16.40	27.88
	12	17.66	29.85	17.41	29.41	17.03	29.14	18.41	31.08	17.26	29.32	17.31	29.85	16.74	28.46	20.64	34.34	18.41	31.65	17.88	30.16	19.82	32.94
	Avg	15.33	25.96	15.25	25.67	15.04	25.95	16.16	27.02	15.46	26.24	15.61	27.02	14.81	25.31	17.28	28.75	16.61	28.71	15.62	26.58	16.57	27.92
PEMS04	3	18.50	29.69	17.73	28.73	18.32	29.30	18.87	29.83	18.31	29.76	18.29	29.37	17.93	28.83	19.46	30.57	18.86	30.02	19.01	30.13	19.68	31.28
	6	19.74	31.49	18.62	30.13	19.38	30.87	19.92	31.50	19.06	31.38	19.04	30.57	18.93	30.33	20.85	32.62	19.54	31.29	20.72	32.57	21.88	34.68
	12	21.76	34.26	20.06	32.18	21.11	33.25	21.78	34.15	20.20	33.09	20.21	32.59	20.52	32.59	24.26	37.30	20.82	33.22	23.73	36.62	26.29	41.19
	Avg	19.74	31.44	18.62	30.07	19.37	30.79	19.93	31.46	19.00	31.11	19.02	30.66	18.92	30.29	21.12	32.92	19.57	31.26	20.82	32.64	22.17	35.06
PEMS07	3	19.88	32.01	18.62	30.60	19.19	31.05	20.24	32.08	19.21	31.48	19.12	31.22	18.93	30.84	21.91	33.84	20.97	33.29	20.26	32.27	20.43	33.03
	6	21.65	34.91	19.96	33.00	20.60	33.59	21.58	34.63	20.59	34.52	20.48	33.95	20.42	33.43	24.37	37.59	22.22	35.79	22.43	35.81	22.98	37.28
	12	24.66	39.39	21.97	36.33	22.88	37.34	23.94	38.48	22.61	38.25	22.33	37.31	22.81	37.04	29.23	44.64	24.29	39.41	26.32	41.50	27.76	44.51
	Avg	21.65	34.80	19.91	32.83	20.59	33.50	21.54	34.48	20.50	34.22	20.39	33.72	20.40	33.25	24.49	37.73	22.20	35.66	22.50	35.77	23.18	37.43
PEMS08	3	14.50	23.19	13.44	21.63	13.99	22.17	14.71	22.94	14.41	22.65	14.42	22.84	13.85	21.94	15.95	24.50	15.27	23.65	14.32	22.55	14.91	23.67
	6	15.80	25.50	14.36	23.44	14.92	23.92	15.62	24.73	15.30	24.65	15.34	24.91	14.79	23.70	17.54	26.96	15.99	25.23	15.42	24.56	16.39	26.53
	12	18.13	28.88	15.80	25.80	16.55	26.42	17.17	27.12	16.67	27.07	16.56	27.34	16.30	26.17	20.70	31.28	17.37	27.55	17.34	27.55	19.26	31.14
	Avg	15.85	25.41	14.37	23.33	14.95	23.83	15.59	24.55	15.26	24.47	15.30	24.80	14.80	23.62	17.64	26.96	16.03	25.18	15.46	24.50	16.52	26.57
PEMS-BAY	3	1.31	2.79	1.32	2.80	1.29	2.71	1.34	2.79	1.34	2.83	1.35	2.85	1.31	2.74	1.46	3.02	1.47	3.02	1.35	2.83	1.39	3.00
	6	1.61	3.67	1.65	3.73	1.61	3.65	1.67	3.72	1.66	3.80	1.67	3.81	1.64	3.70	1.90	4.07	1.80	3.93	1.70	3.85	1.83	4.19
	12	1.88	4.33	1.93	4.43	1.90	4.38	1.94	4.43	1.94	4.47	1.94	4.49	1.95	4.48	2.38	5.11	2.12	4.74	2.05	4.73	2.33	5.35
	Avg	1.55	3.46	1.58	3.52	1.55	3.46	1.60	3.52	1.59	3.57	1.61	3.59	1.58	3.51	1.84	3.91	1.75	3.77	1.64	3.67	1.78	4.01
METR-LA	3	2.80	5.47	2.82	5.55	2.63	5.04	2.83	5.49	2.80	5.51	2.87	5.59	2.70	5.18	3.04	5.71	2.82	5.48	2.81	5.35	2.97	5.86
	6	3.13	6.44	3.18	6.57	2.96	6.02	3.20	6.51	3.17	6.54	3.23	6.61	3.08	6.19	3.50	6.78	3.19	6.50	3.27	6.53	3.55	7.20
	12	3.49	7.34	3.57	7.54	3.37	7.10	3.59	7.48	3.56	7.50	3.59	7.55	3.50	7.25	4.09	8.00	3.61	7.54	3.91	7.94	4.36	8.82
	Avg	3.08	6.26	3.13	6.39	2.93	5.90	3.14	6.33	3.11	6.35	3.17	6.43	3.04	6.05	3.46	6.64	3.15	6.35	3.25	6.43	3.54	7.08

From DCRNN **(2017)** – STAEformer **(2023)**, average ↓ MAE in a ***fair*** **comparison** is **only 1~2%**, what happened ?

From 2017-2023, it was party time of STGNN

Now, the party is over ? Yes, but not

There is still a lot of work to be explored in many stages, including PT / SFT / ICL / TTT, etc.