Lightweight and Interpretable Transformer via Mixed Graph Algorithm Unrolling for Traffic Forecast

Ji Qi[1] Mingxiao Liu[1] Tam Thuc Do[2] Yuzhe Li[1] Zhuoshi Pan[1]
Gene Cheung[2*] H. Vicky Zhao[1*]

[1]Department of Automation, Tsinghua University, Beijing, China
[2]Department of EECS, York University, Toronto, Canada

1 June, 2026

Challenges in Traffic forecasting: modeling complex spatiotemporal dependencies, limited training data, edge device deployment, etc.

Conventional Transformers

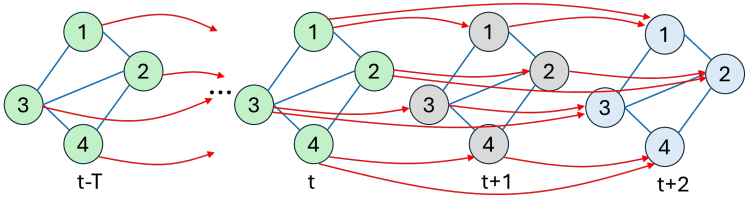
- Data-driven approach with complex dependencies modeled by attention
- Powerful performance
- Large model size and computational cost, lack of interpretability

Graph Signal Processing (GSP)

- Model-based approach by smoothing signals on graphs
- Strong interpretability
- Limited performance and flexibility

Motivation: Building Lightweight and Interpretable Transformers

- *Parameterize* the GSP-based approach for data-driven learning while preserving the interpretability — **Algorithm Unrolling**
- Preserve the *transformer-like architecture* by periodically introducing graph learning modules to learn graphs from data



- **Directed graphs $\mathcal{G}^d$:** temporal dependencies on each node with its previous W states
- **Undirected graphs $\mathcal{G}^u$:** spatial dependencies between nodes with road connections on each timestamp

Mixed Graph Smoothing Priors

- Undirected graph: GLR $= \mathbf{x}^\top \mathbf{L}^u \mathbf{x}$
- Directed graph: DGLR $= \mathbf{x}^\top (\mathbf{L}_r^d)^\top \mathbf{L}_r^d \mathbf{x} \triangleq \mathbf{x}^\top \mathcal{L}_r^d \mathbf{x}$ and DGTV $= \|\mathbf{L}_r^d \mathbf{x}\|_1$

Smoothing the signals on the mixed graph:

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2+\mu_u\mathbf{x}^\top\mathbf{L}^u\mathbf{x}+\mu_{d,2}\mathbf{x}^\top\mathcal{L}_r^d\mathbf{x}+\mu_{d,1}\|\mathbf{L}_r^d\mathbf{x}\|_1. \tag{1}$$

Introduce the auxiliary variables: $\boldsymbol{\phi}=\mathbf{L}_r^d\mathbf{x}$ for the ℓ_1 term, split the ℓ_2 terms and add auxiliary variables $\mathbf{z}_u,\mathbf{z}_d$

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}_u,\mathbf{z}_d}\quad & \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2+\mu_u\mathbf{z}_u^\top\mathbf{L}^u\mathbf{z}_u+\mu_{d,2}\mathbf{z}_d^\top\mathcal{L}_r^d\mathbf{z}_d+(\boldsymbol{\gamma}^\tau)^\top(\boldsymbol{\phi}^\tau-\mathbf{L}_r^d\mathbf{x})+\frac{\rho}{2}\|\boldsymbol{\phi}^\tau-\mathbf{L}_r^d\mathbf{x}\|_2^2\\ \text{s.t.}\quad & \mathbf{x}=\mathbf{z}_u=\mathbf{z}_d,\quad \boldsymbol{\phi}=\mathbf{L}_r^d\mathbf{x}.\end{aligned} \tag{2}$$

Write the augmented Lagrangian function, and solve with ADMM:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}_u,\mathbf{z}_d,\boldsymbol{\phi}}\ & \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2+\mu_u\mathbf{z}_u^\top\mathbf{L}^u\mathbf{z}_u+\mu_{d,2}\mathbf{z}_d^\top\mathcal{L}_r^d\mathbf{z}_d+(\boldsymbol{\gamma}^\tau)^\top(\boldsymbol{\phi}^\tau-\mathbf{L}_r^d\mathbf{x})+\frac{\rho}{2}\|\boldsymbol{\phi}^\tau-\mathbf{L}_r^d\mathbf{x}\|_2^2\\ &+(\boldsymbol{\gamma}_u^\tau)^\top(\mathbf{x}-\mathbf{z}_u)+\frac{\rho_u}{2}\|\mathbf{x}-\mathbf{z}_u\|_2^2+(\boldsymbol{\gamma}_d^\tau)^\top(\mathbf{x}-\mathbf{z}_d)+\frac{\rho_d}{2}\|\mathbf{x}-\mathbf{z}_d\|_2^2+\mu_{d,1}\|\boldsymbol{\phi}^\tau\|_1\end{aligned} \tag{3}$$

1. **Optimize $\mathbf{x}^{\tau+1}$ and auxiliary variables $\mathbf{z}_u^{\tau+1}, \mathbf{z}_d^{\tau+1}$:**
Solve the following linear systems by conjugated gradient (CG) descent:

$$\left(\mathbf{H}^\top\mathbf{H} + \tfrac{\rho}{2}\mathcal{L}_r^d + \tfrac{\rho_u+\rho_d}{2}\mathbf{I}\right)\mathbf{x}^{\tau+1} = (\mathbf{L}_r^d)^\top(\tfrac{\boldsymbol{\gamma}^\tau}{2} + \tfrac{\rho}{2}\boldsymbol{\phi}^\tau) - \tfrac{\boldsymbol{\gamma}_u^\tau}{2} + \tfrac{\rho_u}{2}\mathbf{z}_u^\tau - \tfrac{\boldsymbol{\gamma}_d^\tau}{2} + \tfrac{\rho_d}{2}\mathbf{z}_d^\tau + \mathbf{H}^\top\mathbf{y}, \quad (4)$$

$$\left(\mu_u\mathbf{L}^u + \tfrac{\rho_u}{2}\mathbf{I}\right)\mathbf{z}_u^{\tau+1} = \tfrac{\boldsymbol{\gamma}_u^\tau}{2} + \tfrac{\rho_u}{2}\mathbf{x}^{\tau+1}, \quad (5)$$

$$\left(\mu_{d,2}\mathcal{L}_r^d + \tfrac{\rho_d}{2}\mathbf{I}\right)\mathbf{z}_d^{\tau+1} = \tfrac{\boldsymbol{\gamma}_d^\tau}{2} + \tfrac{\rho_d}{2}\mathbf{x}^{\tau+1}. \quad (6)$$

Structural Interpretation: Alternative *low-pass filtering* with undirected graph kernel $\mathbf{L}^d$ and directed graph kernels $\mathbf{L}_r^d, \mathcal{L}_r^d$

$$\mathbf{x}^{\tau+1} = \mathbf{U}\mathrm{diag}\left(\left\{\tfrac{2}{\rho\xi_k+\rho_u+\rho_d+2\cdot\not{1}_{\mathrm{obs}}}\right\}\right)\mathbf{U}^\top\mathrm{RHS}_{(4)},$$

$$\mathbf{z}_u^{\tau+1} = \mathbf{V}\mathrm{diag}\left(\left\{\tfrac{2}{2\mu_u\lambda_k+\rho_u}\right\}\right)\mathbf{V}^\top\left(\tfrac{\boldsymbol{\gamma}_u^\tau}{2} + \tfrac{\rho_u}{2}\mathbf{x}^{\tau+1}\right),$$

$$\mathbf{z}_d^{\tau+1} = \mathbf{U}\mathrm{diag}\left(\left\{\tfrac{2}{2\mu_{d,2}\xi_k+\rho_d}\right\}\right)\mathbf{U}^\top\left(\tfrac{\boldsymbol{\gamma}_d^\tau}{2} + \tfrac{\rho_d}{2}\mathbf{x}^{\tau+1}\right)$$

2. **Optimize $\boldsymbol{\phi}^{(t+1)}$:** The $\ell_2 - \ell_1$ problem gives a closed-form solution, where S is the soft-thresholding operator:

$$\delta = (\mathbf{L}_r^d)_{i\cdot}\mathbf{x}^{\tau+1} - \frac{1}{\rho}\boldsymbol{\gamma}_i^{\tau}, \quad \phi_i^{\tau+1} = \text{sign}(\delta)\cdot\max(|\delta| - \rho^{-1}\mu_{d,1},\, 0) \triangleq S_{\mu_{d,1}/\rho}(\delta). \tag{7}$$

Structural Interpretation: the high-frequency components $\mathbf{L}_r^d\mathbf{x}$ of signal $\mathbf{x}$ is *attenuated* by the soft-thresholding operation — *low-pass filtering* of signal $\mathbf{x}$ with the directed graph kernel $\mathbf{L}_r^d$ under *filter-bank explanation*

3. **Lagrangian Multipliers $\boldsymbol{\gamma}, \boldsymbol{\gamma}_u, \boldsymbol{\gamma}_d$ update:**

$$\begin{aligned}
\boldsymbol{\gamma}^{\tau+1} &= \boldsymbol{\gamma}^{\tau} + \rho(\boldsymbol{\phi}^{\tau+1} - \mathbf{L}_r^d\mathbf{x}^{\tau+1}),\\
\boldsymbol{\gamma}_u^{\tau+1} &= \boldsymbol{\gamma}_u^{\tau} + \rho_u(\mathbf{x}^{\tau+1} - \mathbf{z}_u^{\tau+1}),\\
\boldsymbol{\gamma}_d^{\tau+1} &= \boldsymbol{\gamma}_d^{\tau} + \rho_d(\mathbf{x}^{\tau+1} - \mathbf{z}_d^{\tau+1})
\end{aligned} \tag{8}$$

Unrolling the ADMM algorithm yields an ADMM block interpreted as **a sequence of learnable low-pass filters** over the mixed graph $\Longrightarrow$ **FFN** in transformer.

Conventional Attention Mechanism:

$$e(i,j) = (\mathbf{Q}\mathbf{x}_j)^\top(\mathbf{K}\mathbf{x}_i) = \mathbf{x}_j^\top(\mathbf{Q}^\top\mathbf{K})\mathbf{x}_i, \quad a_{i,j} = \frac{\exp(e(i,j))}{\sum_{l=1}^{N}\exp(e(i,l))}, \quad \mathbf{y}_i = \sum_{l=1}^{N} a_{i,l}\mathbf{x}_l\mathbf{V} \quad (9)$$

Lightweight Graph Learning Module

- Feature extraction: leveraging neighboring node signals and position information, with learnable function $f(\cdot)$:

$$\mathbf{f}_i = f(i|\mathbf{X}_{\mathcal{N}_i}, \mathbf{E}_{\mathcal{N}_i}) \quad (10)$$

- Graph construction: Gaussian kernel under Mahalanobis distances, with learnable matrices $\mathbf{M}$, $\mathbf{P}$:

$$\mathrm{d}^u(i,j) = (\mathbf{f}_i^u - \mathbf{f}_j^u)^\top\mathbf{M}(\mathbf{f}_i^u - \mathbf{f}_j^u), \quad \mathrm{d}^d(i,j) = (\mathbf{f}_i^d - \mathbf{f}_j^d)^\top\mathbf{P}(\mathbf{f}_i^d - \mathbf{f}_j^d) \quad (11)$$

- Normalization:

$$w_{i,j}^u = \frac{\exp(-\mathrm{d}^u(i,j))}{\sqrt{\sum_{l\in\mathcal{N}_i}\exp(-\mathrm{d}^u(i,l))}\sqrt{\sum_{k\in\mathcal{N}_j}\exp(-\mathrm{d}^u(k,j))}}, \quad w_{j,i}^d = \frac{\exp(-\mathrm{d}^d(i,j))}{\sum_{[k,j]\in\mathcal{E}^d}\exp(-\mathrm{d}^d(k,j))}. \quad (12)$$

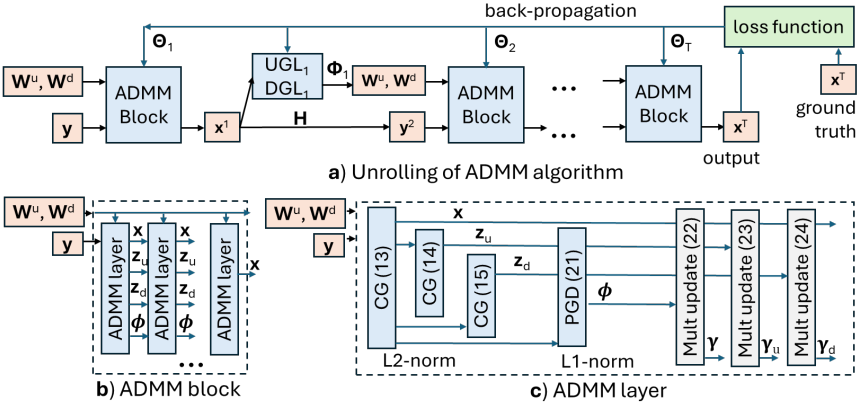


Figure 2: Unrolling of proposed iterative ADMM algorithm into neural layers.

Loss function: Huber loss on the *entire* output sequence $\hat{\mathbf{X}}_{t-T:t+S}$ for the signal recovery problem.

- **Tasks:** 30 / 60 / 120-minute traffic forecasting on real world datasets: PEMS03 (358 nodes, 547 edges) and METR-LA (207 nodes, 1,315 edges)
- **Evaluation metrics:** MAE, RMSE, MAPE (%)
- **Baselines (with parameter count comparison) :**

Category	Selected Models	Params #
Model-based	VAR (2003)	-
GNNs	STGCN (2018), STSGCN (2020)	321K, 3,496K
GATs	GMAN (2020), ST-Wave (2023)	210K, 883K
Transformers	PDFormer (2023), STAEformer (2023), PatchSTG (2025)	531K, 1,404K, 2,283K
Adaptive GNN	Graph WaveNet (2019), AGCRN (2020)	277K, 749K
MLP-based models	STID (2022), SimpleTM (2025)	123K, 540K
Mixed-graph unrolling	Ours	**38K**

Our model employes drastically fewer parameters than all baselines (*7.2% of PDFormer*).

Our model achieves *competitive performance* (top 3 in 4 out of 6 runs) with *significantly fewer parameters*.

Table 1: Comparison of metrics of our lightweight transformer to the baselines. We use **boldface** for the smallest error, color the 2nd and 3nd small error in blue, and underline the next 2 smallest errors.

Dataset & Horizon	PEMS03 (358 nodes, 547 edges)			METR-LA (207 nodes, 1,315 edges)		
	30 minutes	60 minutes	120 minutes	30 minutes	60 minutes	120 minutes
VAR	28.07 / 16.53 / 17.49	30.54 / 18.31 / 19.61	36.65 / 22.40 / 24.64	10.72 / 5.55 / 11.29	12.59 / 6.99 / 13.46	14.83 / 8.90 / 16.54
STGCN	28.06 / 18.24 / 17.76	37.31 / 24.29 / 23.00	54.34 / 34.83 / 30.52	11.26 / 5.08 / 10.93	13.91 / 6.35 / 13.67	16.92 / 8.11 / 17.69
STSGCN	26.41 / 16.75 / 16.86	30.62 / 19.35 / 19.15	38.04 / 23.86 / 23.62	10.25 / 4.05 / 9.18	12.65 / 5.18 / 11.48	15.74 / 6.84 / 15.33
GMAN	25.79 / 16.23 / 20.04	27.57 / 17.48 / 24.33	30.69 / 19.20 / 26.69	11.97 / 5.34 / 10.66	14.49 / 6.93 / 13.12	15.53 / 7.59 / 14.70
ST-Wave	25.57 / 15.11 / 15.04	28.65 / 16.81 / 19.24	**29.88** / **17.11** / 17.71	10.81 / 4.11 / 9.16	13.24 / 5.33 / 11.32	23.18 / 11.22 / 12.71
PDFormer	23.71 / 15.05 / 18.16	27.16 / 17.26 / 21.21	35.77 / 22.25 / 25.01	10.21 / 3.89 / 8.50	12.17 / 4.81 / 10.92	17.27 / 9.55 / 19.10
STAEformer	30.22 / 18.85 / 26.62	38.36 / 23.68 / 29.21	48.72 / 31.96 / 44.71	10.16 / **3.73** / 8.25	12.58 / **4.79** / 10.08	**14.63** / **5.82** / **11.87**
PatchSTG	**22.89** / 14.66 / 17.51	**25.09** / **15.76** / 17.02	29.97 / 18.65 / 21.13	10.24 / 3.89 / 9.02	14.12 / 5.80 / 12.08	17.39 / 7.73 / 16.48
GraphWaveNet	25.76 / 15.22 / 16.83	28.15 / 18.11 / 17.56	34.60 / 21.10 / 22.45	11.42 / 5.18 / 8.21	12.46 / 5.17 / 11.97	23.13 / 11.32 / 13.53
AGCRN	27.40 / 15.19 / **14.35**	29.90 / 16.76 / **15.32**	32.79 / 18.72 / **17.03**	10.11 / 3.82 / 8.61	12.56 / 5.00 / 11.25	14.77 / 6.33 / 14.05
STID	26.50 / 17.27 / 18.59	31.88 / 20.82 / 24.89	42.98 / 28.03 / 42.41	10.21 / 4.05 / 9.47	12.61 / 5.21 / 12.22	15.48 / 6.98 / 16.78
SimpleTM	23.75 / 15.13 / 15.59	25.56 / 15.97 / 15.49	30.58 / 18.73 / 18.18	**8.76** / 4.12 / **7.63**	**11.96** / 5.49 / **9.65**	15.44 / 7.64 / 12.27
Ours	25.05 / 15.85 / 16.49	26.96 / 16.58 / 18.07	34.06 / 20.23 / 23.86	10.06 / 4.05 / 9.23	12.17 / 5.27 / 11.78	15.19 / 7.14 / 16.44

Our model sits distinctly in the lower-left corner in the error-size trade-off plot, achieving the accuracy of models over 10× larger.

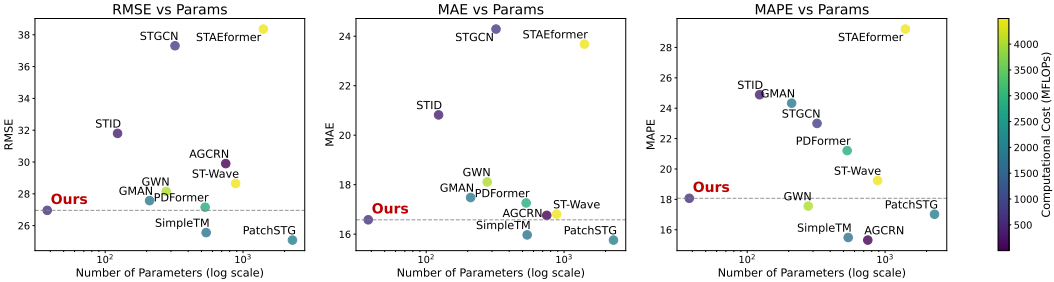


Figure 3: The trade-off between the number of parameters and performance for 60-minute forecast on PEMS03 dataset.

Conclusion & Broader Impacts

- We unroll a mixed-graph smoothing algorithm into a lightweight and interpretable transformer architecture for traffic forecasting, achieving competitive performance with significantly fewer parameters.
- **Broader impacts:**
  - Transformers can be *whitened* and *reduced* as an unrolled network of graph smoothing algorithms.
  - We can learn *directed* attention scores with directed graph modeling, which enhance the generalizability by capturing assymmetrical or causal dependencies within data.

Thanks!