COMPOSING PRIORS FOR TEMPORAL GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal graph models often perform well on individual datasets yet struggle to generalize the same architecture across multiple datasets or domains. We argue that this transfer gap stems from a common architectural choice: jointly learning temporal dynamics and structural propagation entangles domain-specific spatio-temporal dependencies. We show that temporal and structural priors can instead be effectively composed before graph-specific learning. Building on this insight, we introduce a framework that decouples temporal and structural representation learning and equips a GNN with three complementary, domain-agnostic priors: foundation-model temporal embeddings, local temporal-context summaries, and supra-Laplacian positional encodings of the evolving topology. Rather than asking a recurrent architecture to rediscover temporal dynamics and structural propagation for each domain, our framework learns how reusable representations interact over the graph. Across multiple real-world temporal graph benchmarks, it achieves competitive or state-of-the-art supervised performance. On the MiNT zero-shot benchmark, our framework consistently mitigates negative transfer and outperforms strong temporal graph baselines across 20 unseen networks.

1 INTRODUCTION

Many real-world systems are inherently dynamic, comprising time-evolving features and structure (Kazemi et al., 2020; Longa et al., 2023; Gravina & Bacciu, 2024; Jin et al., 2024). Examples include transportation networks with evolving traffic patterns (Li et al., 2018), epidemic spreading over dynamic contact networks (Rozemberczki et al., 2021), and emerging autonomous agentic AI systems operating in complex environments (Kurenkov et al., 2023; Bei et al., 2025). Learning on such temporal graphs therefore requires capturing both temporal dependencies in node observations and the spatial dependencies induced by the evolving topology.

A central challenge in temporal graph learning is that these two sources of information are typically modeled jointly. Existing approaches commonly combine sequence models with Graph Neural Networks (GNNs) (Seo et al., 2018; Li et al., 2018; Zhao et al., 2020; Bai et al., 2021; Cini et al., 2023; De Castelli et al., 2026), or incorporate graph convolutions directly into recurrent architectures (Chen et al., 2022; Li et al., 2019b; Micheli & Tortorella, 2022). These architectures couple temporal modeling with graph propagation, which constrains the temporal representations to be learned together with the structural patterns of the graph. This coupling also introduces a sequential dependency during training, requiring Backpropagation Through Time (BPTT) (Rumelhart et al., 1986) and thus increasing the computational burden. Moreover, it can hinder transfer to graphs whose topology, dynamics, or node identities differ from those observed during training. Recent transfer-learning analyses on dynamic interaction networks, for instance, show that memory-based architectures can fail catastrophically on entirely unseen graphs because their recurrent components memorize train-time node identities (Chatterjee et al., 2025).

We address these challenges by separating temporal representation extraction from structural graph learning. Instead of jointly learning temporal and structural dependencies within a recurrent architecture, our framework constructs temporal and structural representations independently and composes them at the graph learning stage. This enables temporal information to be extracted independently of the target topology, while the GNN focuses on modeling how these representations interact according to the graph structure.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We introduce **MOSAIC** (**MO**dular **S**tructural **A**nd temporal Prior **I**ntegration by **C**omposition), a modular framework that combines three complementary priors: temporal representations obtained from pre-trained Time-Series Foundation Models (TSFMs) (Goswami et al., 2024; Ansari et al., 2024), summaries of local temporal context, and supra-Laplacian positional encodings (Galron et al., 2026a) of the evolving topology. The use of supra-Laplacian as structural prior provides a natural way to couple graph geometry across consecutive snapshots and avoid treating snapshot-wise positional representations independently. The resulting representations are treated as fixed inputs to a static GNN, which learns their interactions across the graph. By disentangling structural and temporal dynamics, our method can learn to compose them and learn patterns that are transferable to unseen graphs. Moreover, we decrease the training cost because we do not use BPTT.

Our main contributions are as follows: *(i)* We propose MOSAIC, a modular framework that separates temporal representation extraction from structural graph learning. It combines foundation-model temporal embeddings, local temporal summaries, and supra-Laplacian positional encodings as complementary priors, which are subsequently composed by a static GNN. *(ii)* We show that this decomposition provides competitive or state-of-the-art performance across multiple supervised temporal graph learning benchmarks, demonstrating that the proposed architecture can effectively capture temporal graph dynamics without jointly training a recurrent temporal encoder and a GNN. *(iii)* We evaluate MOSAIC in zero-shot transfer across 20 unseen networks on the MiNT benchmark (Shamsi et al., 2026), where it consistently reduces negative transfer compared with strong memory-based temporal graph baselines.

2 Related Work

Spatio-Temporal Foundation Models. Recent advancements in spatio-temporal forecasting have sought to generalize predictions across unseen geographic regions. OpenCity (Li et al., 2025) achieves zero-shot transfer for urban traffic forecasting by leveraging Reversible Instance Normalization (Kim et al., 2022) over traffic regions and temporal patch embeddings. Similarly, frameworks such as ST-LLM (Liu et al., 2024) and UniFlow (Yuan et al., 2025) align raster and graph-based spatio-temporal data by leveraging the cognitive routing of Large Language Models (LLMs) (Brown et al., 2020) or shared memory banks for urban patterns. However, these architectures are intrinsically tied to specific geographic grids and regression tasks, and they assume a time-invariant underlying topology. In contrast, MOSAIC operates on general temporal graphs with evolving topology and is not restricted to regression-based urban forecasting tasks.

Transfer Learning on Dynamic Interaction Networks. Temporal graph learning on pure interaction networks has historically struggled with transferability due to a memory bottleneck: standard dynamic graph architectures (Gravina & Bacciu, 2024) rely on recurrent modules (Rumelhart et al., 1986; Cho et al., 2014; Hochreiter & Schmidhuber, 1997) that memorize train-time node identities, and fail catastrophically on entirely unseen graphs (Chatterjee et al., 2025). Recent efforts bypass this limitation through multi-network pre-training. MiNT (Shamsi et al., 2026) demonstrates zero-shot transfer across temporal transaction networks by shuffling input orders and context-switching across 64 datasets to prevent identity leakage. In the domain of Temporal Knowledge Graphs (TKGs), POSTRA (Pan et al., 2025) achieves inductive transfer by abandoning absolute timestamps in favor of relative temporal signals and purely topological relation embeddings. Our method, instead, relies on the computation of strong temporal and structural priors, separating the temporal modeling from graph propagation.

Time-Series Foundation Models as Temporal Priors. Recent time-series research has rapidly consolidated around universal foundation models. Architectures such as MomentFM (Goswami et al., 2024), Chronos (Ansari et al., 2024), Moirai (Woo et al., 2024), FlowState (Graf et al., 2026), and PatchTST (Nie et al., 2023) demonstrate that massive pre-training on diverse sequential data yields robust, domain-agnostic representations of temporal dynamics. By treating time windows as continuous patches or quantizing them into discrete tokens, these models remove the need for dataset-specific temporal resolutions. However, they operate on temporal sequences alone and do not model relational structure or interactions between entities. In contrast, MOSAIC combines their domain-agnostic temporal representations with positional and structural encodings, and leverages a GNN to model information propagation over space.

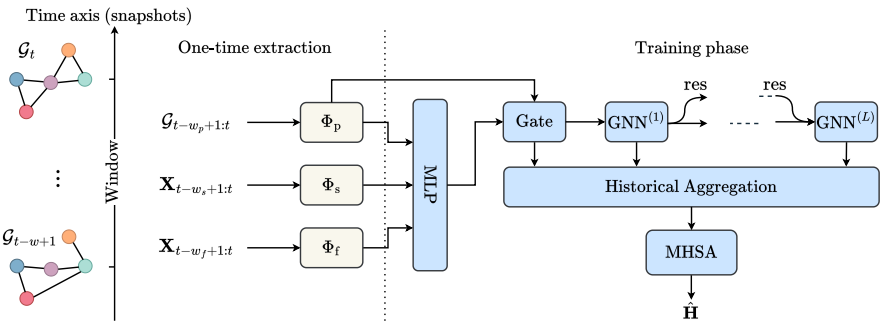

Figure 1: Overall architecture of MOSAIC. On the left, a high-level block architecture showing components across space and time. On the right, the trainable architecture, which consists of the projections, L GNN layers, and historical aggregations of latent states.

3 Composing Priors for Temporal Graph Learning

In this section, we introduce **MOSAIC** (**MO**dular **S**tructural **A**nd temporal Prior **I**ntegration by **C**omposition), a unified framework for encoding temporal graph data as a composition of structured operators. Rather than training a spatio-temporal architecture from scratch, we decompose representation learning into complementary components, each introducing a distinct inductive bias. Our approach follows the time-then-graph paradigm (Gao & Ribeiro, 2022), which has been shown to be advantageous for expressivity compared to entangled paradigms such as time-and-graph. In Figure 1, we provide a visual overview of the components of MOSAIC.

3.1 Notation and Preliminaries

We focus on Discrete-Time Dynamic Graph (D-TDG) (Kazemi et al., 2020; Gravina & Bacciu, 2024), also known as Snapshot-based Temporal Graphs (Longa et al., 2023), where the graph evolves over time through a sequence of snapshots with evolving node features. We define a D-TDG as a sequence of graph snapshots $\mathcal{G}_{1:T} = \{\mathcal{G}_1, \dots, \mathcal{G}_T\}$. Each snapshot $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$ consists of a non-empty set of nodes $\mathcal{V}_t$, an edge set $\mathcal{E}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ and a node feature matrix $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{V}_t| \times d_n}$. The structural information expressed by $\mathcal{E}_t$ is encoded in the adjacency matrix $\mathbf{A}_t \in \{0,1\}^{n \times n}$, where $n = |\mathcal{V}_t|$ and $[\mathbf{A}_t]_{uv} = 1$ if $(u, v) \in \mathcal{E}_t$, and 0 otherwise. We note that when $\mathbf{A}_t = \mathbf{A}_1 \; \forall t$ the D-TDG is often called spatio-temporal graph. Finally, we define the degree matrix as $\mathbf{D}_t = \mathrm{diag}(\mathbf{A}_t \mathbf{1})$, and the ordered sequence of input signals as $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_T) \in \mathbb{R}^{T \times n \times d_n}$.

3.2 Time-Signal-Topology Decoupling

We formulate our framework as a composition of three components, in which temporal signals, structural topology, and positional information are encoded independently and then integrated into a unified representation space. This design is formalized through a representation map $\Phi : \mathbb{R}^{T \times n \times d_n} \times \mathfrak{G}^T \to \mathbb{R}^{T' \times n \times d}$ that produces node-wise features suitable for downstream prediction tasks, where $\mathfrak{G}$ denotes the space of graphs over $\mathcal{V}$ and $T' \leq T$ accounts for context-window boundary effects. We define Φ as the concatenation of three complementary operators:

$$\Phi(\mathcal{X}, \mathcal{G}_{1:T}) \;=\; \big[\, \Phi_{\mathrm{f}}(\mathcal{X}) \,\big\|\, \Phi_{\mathrm{s}}(\mathcal{X}) \,\big\|\, \Phi_{\mathrm{p}}(\mathcal{G}_{1:T}) \,\big], \tag{1}$$

where Φ_{f} extracts transferable temporal representations from a frozen pre-trained sequence model, Φ_{s} captures statistical structure from local temporal contexts, and Φ_{p} injects positional and structural information from the evolving graph topology. The decomposition partitions the representation space as $\mathbb{R}^d = \mathcal{H}_{\mathrm{f}} \oplus \mathcal{H}_{\mathrm{s}} \oplus \mathcal{H}_{\mathrm{p}}$, with each subspace carrying one of these three sources of variation. Crucially, the parameters of Φ_{f} are frozen, while Φ_{s} and Φ_{p} have no parameters. This design allows us to learn from features that encode patterns of the underlying graph signal, and allows for a BPTT-free training.

Temporal signal encoding. The operator Φ_{f} maps the temporal signal to a sequence of node-wise embeddings via a pre-trained sequence model f_θ, with frozen set of parameters θ, applied over a

sliding context of length w_f, so that $\Phi_{\mathrm{f}}(\mathcal{X})_t = f_\theta(\mathcal{X}_{t-w_f+1:t}) \in \mathbb{R}^{n\times d_f}$. Here, f_θ acts independently on each node trajectory, modeling temporal dependencies while ignoring the underlying graph structure. This lets Φ_{f} leverage transferable, sequence-level representations learned from large-scale pre-training, while remaining agnostic to spatial interactions.

Local statistical summaries. The operator Φ_{s} complements Φ_{f} by extracting a localized representation of node features over a possibly different temporal context w_s. By either directly preserving or aggregating the recent signal, this component introduces a controlled inductive bias that grounds the deep temporal representation of Φ_{f} in the immediate state of the network.

Supra-Laplacian positional encoding. We define Φ_{p} to inject structural and temporal priors via the supra-Laplacian (Karmim et al., 2024) of a sliding topology window. While standard Laplacian positional encodings (Dwivedi & Bresson, 2021) are highly effective for static graphs, computing them independently per snapshot suffers from temporal misalignment and generalizes worse (Galron et al., 2026a). Therefore, given a window of length w_p ending at time t, we merge the snapshots $\mathcal{G}_{t-w_p+1:t}$ into a multilayer graph (Kivelä et al., 2014) and assemble the supra-Laplacian matrix

$$\mathbf{L}_{\mathrm{sup}} \;=\; \mathbf{I} - \mathbf{D}_{\mathrm{sup}}^{-\frac{1}{2}}\mathbf{A}_{\mathrm{sup}}\mathbf{D}_{\mathrm{sup}}^{-\frac{1}{2}} \;\in\; \mathbb{R}^{nw_p\times nw_p}, \tag{2}$$

where the supra-adjacency $\mathbf{A}_{\mathrm{sup}}$ is block tridiagonal with diagonal blocks $[\mathbf{A}_{\mathrm{sup}}]_{\tau\tau} = \mathbf{A}_\tau$ for $\tau \in [t-w_p+1, t]$ and off-diagonal blocks $[\mathbf{A}_{\mathrm{sup}}]_{\tau,\tau\pm1} = \mu\mathbf{I}_n$, with $\mu > 0$ controlling the inter-layer coupling and all other blocks zero. $\mathbf{D}_{\mathrm{sup}}$ is the degree matrix associated with $\mathbf{A}_{\mathrm{sup}}$. Being constructed from a multilayer graph, $\mathbf{L}_{\mathrm{sup}}$ preserves the standard properties of a graph Laplacian (Chung, 1997). The k smallest eigenpairs $\{(\lambda_j, \mathbf{u}_j)\}_{j=1}^{k}$ of $\mathbf{L}_{\mathrm{sup}}$ correspond to the minimizers of the variational objective

$$\min_{\mathbf{V}^\top\mathbf{D}_{\mathrm{sup}}\mathbf{V}=\mathbf{I}} \sum_{\tau=t-w_p+1}^{t} \mathrm{tr}\big(\mathbf{V}_\tau^\top(\mathbf{D}_\tau - \mathbf{A}_\tau)\mathbf{V}_\tau\big) + \mu \sum_{\tau=t-w_p+2}^{t} \|\mathbf{V}_\tau - \mathbf{V}_{\tau-1}\|_F^2\,, \tag{3}$$

where $\mathbf{V} = [\mathbf{V}_{t-w_p+1}^\top, \ldots, \mathbf{V}_t^\top]^\top \in \mathbb{R}^{nw_p\times k}$ stacks the per-snapshot blocks, and $\mathbf{U} = \mathbf{D}_{\mathrm{sup}}^{1/2}\mathbf{V} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$. The first term in Equation (3) enforces intra-layer structural smoothness through the Laplacian quadratic form, while the second term enforces inter-layer (i.e., over time) smoothness of the computed eigenvectors (the proof is in Appendix C). We define the positional encoding at time t by concatenating the eigenvectors with their corresponding eigenvalues, thereby retaining their spectral scales:

$$\Phi_{\mathrm{p}}(\mathcal{G}_{t-w_p+1:t}) \;=\; \big[\,\mathbf{U}_t \,\big\|\, \mathbf{1}_n\boldsymbol{\lambda}^\top\,\big] \;\in\; \mathbb{R}^{n\times 2k}, \tag{4}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)^\top \in \mathbb{R}^k$ collects the corresponding eigenvalues and $\mathbf{1}_n \in \mathbb{R}^n$ broadcasts them across nodes.

Since only the k smallest eigenpairs are required, we avoid the full eigendecomposition and approximate them directly using the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method, which is scalable to large inputs (Knyazev, 2001). By providing the model with a strong inductive bias on the local geometry of the evolving graph, Φ_{p} explicitly grounds the data-driven temporal summaries Φ_{f} and Φ_{s} within a structural context.

Trajectory-Aware Layer Aggregation. The three operator outputs $\{\Phi_{\mathrm{f}}, \Phi_{\mathrm{s}}, \Phi_{\mathrm{p}}\}$ are projected into a shared d-dimensional latent space, additively combined, and passed through a non-linear activation gated by the positional encodings, adaptively modulating the structural bias to the topology of the temporal window. Since sequence modeling is handled in a one-time preprocessing stage, the GNN that models spatial dependencies can be scaled to L layers with residual connections at a reduced training cost compared to recurrent or spatio-temporal architectures. This expands the spatial receptive field without the memory overhead of BPTT.

We model spatial propagation via a GNN over the projected features. Starting from the initial representation $\mathbf{H}_0$, each GNN layer propagates information over the input graph at time t, $\mathcal{G}_t$, with a residual connection:

$$\mathbf{H}_l = \mathbf{H}_{l-1} + \mathrm{GNN}_l(\mathbf{H}_{l-1}, \mathcal{G}_t), \qquad l = 1, \ldots, L, \tag{5}$$

We chose to implement skip-connections in GNN because they improve optimization (Xu et al., 2021) and enable the use of a deeper architecture (Li et al., 2019a).

We enable adaptive spatial receptive fields by retaining and jointly aggregating representations across GNN depths. In message-passing GNNs, the representation at each layer incorporates information from progressively larger neighborhoods, making the network depth directly determine its effective receptive field. However, the appropriate neighborhood range can vary across nodes and tasks, motivating architectures that combine representations from different depths rather than relying exclusively on the deepest layer (Xu et al., 2018). Following this principle, reading out the final hidden state $\mathbf{H}_L$ alone would collapse the multi-scale signal into a single embedding and discard the representations associated with shallower receptive fields. We instead treat the full trajectory $\mathbf{H} = [\mathbf{H}_0 \,\|\, \mathbf{H}_1 \,\|\, \dots \,\|\, \mathbf{H}_L] \in \mathbb{R}^{n\times(L+1)\times d}$ as the model's primary representation, and pass it to the historical activation pooling of Galron et al. (2026b) as a single multi-scale tensor in which all $L+1$ activations are jointly attended to. We also sum sinusoidal positional encodings $\mathbf{P}$ (Vaswani et al., 2017) along the layer axis to make depth ordering explicit, $\tilde{\mathbf{H}} = \mathbf{H} + \mathbf{P}$. Then, we include a single attention head that treats the final-layer state as the query, $\mathbf{Q} = \tilde{\mathbf{H}}_L \mathbf{W}^Q \in \mathbb{R}^{n\times 1\times d}$, while keys $\mathbf{K} = \tilde{\mathbf{H}}\mathbf{W}^K$ and values $\mathbf{V} = \tilde{\mathbf{H}}$, both in $\mathbb{R}^{n\times(L+1)\times d}$, span the entire trajectory. Scaled dot-product scores are then averaged across nodes, producing a single weighting vector

$$c_l \;=\; \frac{1}{n}\sum_{u\in\mathcal{V}} \frac{\mathbf{Q}_u^\top \mathbf{K}_{u,l}}{\sqrt{d}}, \qquad l \in \{0,\dots,L\}, \tag{6}$$

that summarizes how much each depth scale contributes to the final-layer view of the graph. Following Galron et al. (2026b), the weights are normalized as $\alpha_l = c_l / \sum_{l'} c_{l'}$ rather than via softmax. This normalization does not constrain the coefficients to be positive, allowing the aggregation to assign signed weights to different depth scales. Consequently, the pooling can represent differences between layers rather than only convex combinations of their activations:

$$\mathbf{Z} \;=\; \sum_{l=0}^{L} \alpha_l \,\tilde{\mathbf{H}}_l \;\in\; \mathbb{R}^{n\times d} \tag{7}$$

We subsequently apply node-wise Multi-Head Self-Attention (MHSA) (Vaswani et al., 2017) to $\mathbf{Z}$, restoring inter-node interactions after the layer-wise pooling $\hat{\mathbf{H}} = \mathrm{MHSA}(\mathbf{Z}, \mathbf{Z}, \mathbf{Z})$. Coupling residual message passing with trajectory-aware aggregation thus turns the deep GNN into a multi-scale spatial encoder.

Implementation Details. In our experiments in Section 4, we instantiate the components of our framework as follows. *(i)* We set Φ_f to the pre-trained `Moment-1-small` (Goswami et al., 2024), which contains approximately 40M parameters and is used without fine-tuning; the context window w_f is selected according to the task. *(ii)* We define Φ_s to be the identity mapping (i.e., $\Phi_\mathrm{s}(\mathcal{X})_t = \mathcal{X}_{t-w_s+1:t}$), or a fixed-dimensional representation obtained via adaptive average pooling over the temporal and channel dimensions. The choice depends on the task and enforces a controlled inductive bias toward low-variance components of the signal. *(iii)* For spatio-temporal data, we simulate mild topology variations by randomly inserting and removing up to 5% of edges at each training step, improving robustness to temporal variability in the underlying graph. *(iv)* As the trainable backbone, we employ a Graph Convolutional Network (GCN) (Kipf & Welling, 2017) with the trajectory-aware aggregation mechanism described in the previous paragraph.

3.3 Complexity Analysis

The computational cost of MOSAIC is divided into two distinct phases, separating the one-time temporal representation extraction from the subsequent spatial optimization. This separation is central to the computational advantages of our approach: although pre-processing introduces an upfront cost, it is performed once and its outputs can be reused across training runs, while the GNN is optimized without the need of BPTT. To quantify this trade-off, we report the measured pre-processing runtime, training time, and allocated memory of MOSAIC in Appendix B.6.

Preprocessing. Extracting the temporal embeddings requires passing n independent sequences of length w_f through the TSFM, inheriting its inference time and space complexities which for MomentFM correspond to those of the standard Transformer (Goswami et al., 2024). Extracting the supra-Laplacian PE requires computing the k smallest eigenpairs of $\mathbf{L}_\mathrm{sup}$. Using LOBPCG, each iteration requires $\mathcal{O}(k\,\mathrm{nnz}(\mathbf{L}_\mathrm{sup}))$ operations, where $\mathrm{nnz}(\mathbf{L}_\mathrm{sup})$ denotes the number of nonzero entries in the supra-Laplacian. Its space complexity is $\mathcal{O}(\mathrm{nnz}(\mathbf{L}_\mathrm{sup}) + knw_p)$, where the first term

stores the sparse supra-Laplacian and the second stores the k iteratively updated eigenvector estimates. In Appendix B.6.1 we show that the method can be applied to large graphs.

Training phase. During training, the temporal dimension is entirely factored out, circumventing the memory-intensive Backpropagation Through Time (BPTT) of standard sequence models (Rumelhart et al., 1986). The L-layer GNN with residual connections propagates messages at the standard cost $\mathcal{O}(L(|\mathcal{E}| + n))$ (when message-passing based GNNs (Gilmer et al., 2017) are considered), and the node-wise MHSA adds $\mathcal{O}(n^2)$ (Galron et al., 2026b), giving a per-step complexity of $\mathcal{O}(L(|\mathcal{E}| + n) + n^2)$ that is dominated by the spatial attention term in the typical setting $L \ll n$. A step stores only the residual trajectory $\mathbf{H} \in \mathbb{R}^{n\times(L+1)\times d}$; although MHSA has $\mathcal{O}(n^2)$ computational complexity, its fused scaled dot-product attention kernels do not materialize the full $n \times n$ attention map (Rabe & Staats, 2022; Dao et al., 2022), yielding a total memory cost of $\mathcal{O}(nLd)$ per step that depends on none of T, w_f, or w_p: those windows are consumed during pre-processing and no activation is kept alive across time steps. In contrast, end-to-end temporal models incur an additional multiplicative factor proportional to the context length.

4 Experiments

We assess MOSAIC across two settings: *(i)* **Supervised Learning**, where we train the architecture (i.e., only the GNN) under isolated data distributions; and *(ii)* **Zero-Shot Transfer**, pre-training MOSAIC on 64 token networks of the MiNT benchmark (Shamsi et al., 2026) and evaluating it on 20 unseen networks. Implementation details, baselines, and dataset statistics are reported in Appendix A. All experiments were performed on NVIDIA A100-SXM4-40GB GPUs.

4.1 Supervised Node-Level Tasks

We evaluate MOSAIC on 9 supervised node-level tasks. Seven are node-level forecasting benchmarks spanning web activity, public health, social interactions, and traffic forecasting: Wikipedia Math, ChickenPox Hungary, Twitter Tennis (Rozemberczki et al., 2021), and PeMS03/04/07/08 (Song et al., 2020). The last two considered tasks are the node-level classification benchmarks of Li et al. (2024), which are *transductive* tasks that aim to predict the semantic category of each node from its evolving temporal graph context, with 10 and 40 classes for DBLP-10 and arXiv, respectively. We adhere to the data splits and protocols of the original works.

Table 1: Test MSE ↓ on Wikipedia Math, ChickenPox Hungary, and Twitter Tennis, averaged over 10 runs. **First**, **second**, and **third** best results are color-coded. Baseline results are reported from Micheli & Tortorella (2022); Errica et al. (2023).

Model	Wiki Math	Tennis	ChickenPox
DCRNN	$0.679_{\pm 0.007}$	$0.478_{\pm 0.004}$	$1.097_{\pm 0.006}$
GCRN-GRU	$0.680_{\pm 0.021}$	$0.477_{\pm 0.007}$	$1.103_{\pm 0.004}$
GCRN-LSTM	$0.678_{\pm 0.008}$	$0.477_{\pm 0.006}$	$1.097_{\pm 0.006}$
GC-LSTM	$0.677_{\pm 0.009}$	$0.475_{\pm 0.010}$	$1.095_{\pm 0.005}$
DyGrAE	$0.621_{\pm 0.012}$	$0.480_{\pm 0.005}$	$1.102_{\pm 0.013}$
EvolveGCN-H	$0.779_{\pm 0.031}$	$0.481_{\pm 0.003}$	$1.137_{\pm 0.026}$
EvolveGCN-O	$0.807_{\pm 0.047}$	$0.484_{\pm 0.002}$	$1.135_{\pm 0.011}$
A3T-GCN	$0.618_{\pm 0.008}$	$0.477_{\pm 0.005}$	$1.078_{\pm 0.009}$
T-GCN	$0.616_{\pm 0.011}$	$0.478_{\pm 0.004}$	$1.083_{\pm 0.011}$
MPNN LSTM	$0.856_{\pm 0.021}$	$0.482_{\pm 0.001}$	$1.125_{\pm 0.005}$
DynGESN	**$0.610_{\pm 0.003}$**	**$0.300_{\pm 0.003}$**	**$0.907_{\pm 0.007}$**
HMM4G	**$0.542_{\pm 0.008}$**	**$0.333_{\pm 0.004}$**	**$0.939_{\pm 0.013}$**
MOSAIC	**$0.440_{\pm 0.002}$**	**$0.262_{\pm 0.002}$**	**$0.745_{\pm 0.003}$**

Forecasting Tasks. Tables 1 and 2 report forecasting performance. MOSAIC outperforms all baselines on ChickenPox Hungary, Wikipedia Math, and Twitter Tennis. The gains are particularly pronounced on Twitter Tennis (whose topology evolves) and ChickenPox Hungary (whose topology is static), indicating that the combination of pre-computed temporal and structural priors provides a useful representation for forecasting. On the four traffic flow benchmarks, MOSAIC remains competitive with specialized spatio-temporal forecasting models, with errors corresponding to only a small fraction of the traffic volume even on busy roads Chen et al. (2001). We report in Appendix B.3 additional traffic forecasting experiments.

Classification Tasks. Table 3 shows that MOSAIC ranks first in Macro-F1 and second in Micro-F1 on DBLP-10 out of 12 baselines, outperforming ROLAND (You et al., 2022), the second best baseline, by 1.7 points in Micro-F1, and achieves the best Micro-F1 on arXiv with lower variance across runs. Node attributes in arXiv are identical across all snapshots, so the temporal prior receives a time-invariant signal and this benchmark does not exercise the temporal component of MOSAIC.

Table 2: Performance Comparison with Baseline Models on PeMS03, 04, 07, 08. Masked MAE (↓), RMSE (↓) and MAPE (↓, in %) on the 12-step forecasting horizon. **First**, **second**, and **third** best results are color-coded. Baseline results are taken from the respective original papers.

Model	PeMS03			PeMS04			PeMS07			PeMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
TCN	19.31	33.24	19.86	31.11	37.25	15.48	32.68	42.23	14.22	22.69	35.79	14.04
Transformer	17.50	30.24	16.80	23.83	37.19	15.57	26.80	42.95	12.11	18.52	28.68	13.66
DCRNN	18.18	30.31	18.91	24.70	38.12	17.12	25.30	38.58	11.66	17.86	27.83	11.45
STGCN	17.49	30.12	17.15	22.70	35.55	14.59	25.38	38.78	11.08	18.02	27.83	11.40
ASTGCN	17.69	29.66	19.40	22.93	35.22	16.56	28.05	42.57	13.92	18.61	28.16	13.08
GWNet	19.85	32.94	19.31	25.45	39.70	17.29	26.85	42.78	12.12	19.13	31.05	12.68
STSGCN	17.48	29.21	16.78	21.19	33.65	13.90	24.26	39.03	10.21	17.13	26.80	10.96
STFGNN	16.77	28.34	16.30	19.83	31.88	13.02	22.07	35.80	9.21	16.64	26.22	10.60
STGODE	16.50	27.84	16.69	20.84	32.82	13.77	22.99	37.54	10.14	16.81	25.97	10.62
DSTAGNN	15.57	27.21	14.68	19.30	31.46	12.70	21.42	34.51	9.01	15.67	24.77	9.94
ST-WA	15.17	26.63	15.83	**19.06**	31.02	**12.52**	20.74	34.05	8.77	15.41	**24.62**	9.94
ASTGNN	15.07	26.88	15.80	19.26	31.16	12.65	22.23	35.95	9.25	15.98	25.67	9.97
EnhanceNet	16.05	28.33	15.83	20.44	32.37	13.58	21.87	35.57	9.13	16.33	25.46	10.39
AGCRN	16.06	28.49	15.85	19.83	32.26	12.97	21.29	35.12	8.97	15.95	25.22	10.09
Z-GCNETs	16.64	28.15	16.39	19.50	31.61	12.78	21.77	35.17	9.25	15.76	25.11	10.01
STNorm	15.32	25.93	**14.37**	19.21	32.30	13.05	20.59	34.86	8.61	15.39	24.80	**9.91**
STEP	**14.22**	**24.55**	**14.42**	**18.20**	**29.71**	**12.48**	**19.32**	**32.19**	**8.12**	**14.00**	**23.41**	**9.50**
PDFormer	**14.94**	**25.39**	15.82	**18.32**	29.97	**12.10**	**19.83**	32.87	**8.53**	**13.58**	**23.51**	**9.05**
MOSAIC	**14.68**	**22.87**	**14.57**	19.08	**29.55**	13.00	**19.68**	**34.45**	**8.30**	**14.09**	27.51	10.31

Table 3: Node classification on DBLP-10 and arXiv, Micro- and Macro-F1 (↑). **Best** and **second-best** per column are highlighted. Baseline results are reported from Li et al. (2024).

Method	DBLP-10		arXiv	
	Micro-F1 ↑	Macro-F1 ↑	Micro-F1 ↑	Macro-F1 ↑
DeepWalk	66.38	67.12	66.54 ± 0.3	43.01 ± 0.3
Node2Vec	67.31	66.93	67.71 ± 0.5	43.69 ± 0.4
HTNE	68.79	68.36	65.48 ± 0.3	42.25 ± 0.3
M2DNE	69.71	69.75	66.91 ± 0.5	43.52 ± 0.6
DynamicTriad	66.95	66.42	61.10 ± 0.2	38.25 ± 0.3
MPNN	67.74 ± 0.3	65.05 ± 0.5	64.68 ± 1.7	41.22 ± 1.5
STAR	72.98 ± 1.5	72.03 ± 1.2	OOM	OOM
tNodeEmbed	74.19 ± 1.8	74.23 ± 2.2	OOM	OOM
EvolveGCN	71.32 ± 0.5	71.20 ± 0.7	65.17 ± 1.4	43.01 ± 1.3
SpikeNet	74.86 ± 0.5	74.65 ± 0.5	66.69 ± 0.9	43.96 ± 1.0
ROLAND	75.01 ± 1.1	74.98 ± 1.0	68.27 ± 1.2	48.01 ± 1.3
GraphSSM	**76.80 ± 0.3**	**76.00 ± 0.4**	**70.67 ± 0.7**	**49.97 ± 0.5**
MOSAIC	**76.75 ± 0.5**	**76.26 ± 0.4**	**70.82 ± 0.2**	**49.40 ± 0.3**

4.2 Component Ablation

We ablate the three priors to quantify their individual contributions and determine whether the performance of MOSAIC results from combining temporal, local, and structural information. We achieve it by evaluating six ablated variants of MOSAIC on the datasets of Table 1 and on PeMS07, fixing the optimal hyperparameters and dropping one or two of the three priors Φ_f, Φ_s, Φ_p. Table 4 shows three main findings. *(i)* The three priors are complementary: no single prior recovers the full model on its own, and removing any one of them degrades performance on every benchmark, with the single exception of Φ_f on Wikipedia Math, where the leave-one-out variant matches the full model (0.439 against 0.440); *(ii)* local statistics carry the most signal, since removing Φ_s causes the largest leave-one-out drop across all four datasets and Φ_s alone is the strongest single prior on three of the four, which we attribute to the need to provide real inputs to make predictions aware of the local input data scale and variance; and *(iii)* positional encodings are needed regardless of whether the underlying

Table 4: Component ablation on Wikipedia Math, Twitter Tennis, ChickenPox Hungary (10 seeds, MSE ↓) and PeMS07 (5 seeds, MaskedMAE ↓), mean ± std. *w/o* variants drop one prior, *only* variants retain one. Removing Φ_p also bypasses the PE-conditioned gate. **Best** and **second-best** per column are highlighted.

Configuration	Wiki Math MSE ↓	Twitter Tennis MSE ↓	ChickenPox MSE ↓	PeMS07 MaskedMAE ↓
MOSAIC (full)	**0.440 ± 0.002**	**0.262 ± 0.002**	**0.745 ± 0.003**	**19.767 ± 0.055**
w/o TSFM (Φ_f)	**0.439 ± 0.004**	0.293 ± 0.007	**0.839 ± 0.029**	21.043 ± 0.087
w/o Local (Φ_s)	0.669 ± 0.019	0.300 ± 0.003	1.006 ± 0.029	21.202 ± 0.144
w/o Supra-PE (Φ_p)	0.452 ± 0.002	**0.269 ± 0.003**	0.845 ± 0.026	**19.962 ± 0.036**
only TSFM (Φ_f)	0.674 ± 0.016	0.287 ± 0.006	0.971 ± 0.014	21.243 ± 0.028
only Local (Φ_s)	0.471 ± 0.007	0.286 ± 0.001	0.927 ± 0.012	22.483 ± 0.075
only Supra-PE (Φ_p)	0.775 ± 0.007	0.332 ± 0.010	1.098 ± 0.001	45.098 ± 0.002

graph is static or time-varying, but carry little signal in isolation, since only Φ_p collapses on every dataset and most severely on PeMS07. Notably, the largest leave-one-out degradation associated with Φ_p occurs on the static ChickenPox dataset rather than on the dynamically evolving Twitter Tennis network (D-TDG), indicating that its utility is not restricted to time-varying topologies.

Finally, having established that the three priors contribute to the performance of MOSAIC, we examine whether their effectiveness depends on their specific instantiation. For the temporal prior Φ_f, we vary the pretrained backbone to determine whether the observed gains depend on a particular TSFM. As shown in Appendix B.5, replacing MOMENT with Moirai (Woo et al., 2024) and FlowState (Graf et al., 2026) yields strong performance across all seven datasets, with improvements on some datasets, indicating that different foundation models can further improve performance. We perform a similar analysis for the structural prior Φ_p by varying the number of retained eigenvectors k, the inter-layer coupling μ, and the temporal context window w_p. As reported in Appendix B.4, performance remains stable across these configurations, indicating that the supra-Laplacian representation is not highly sensitive to its hyperparameterization.

4.3 Zero-Shot Transfer on MiNT

We evaluate MOSAIC on the 84 Ethereum transaction networks of the MiNT benchmark (Shamsi et al., 2026), comparing it with GC-LSTM and HTGN, the two architectures used in the benchmark for both supervised learning and zero-shot transfer. The task is binary graph property prediction at weekly resolution on featureless interaction graphs, forecasting whether network growth (transaction volume) will increase or decrease over the next week, with constant node features $\mathcal{X} = \mathbf{1}$ for every node and snapshot. We consider two variants. **MOSAIC** is trained in a fully supervised fashion on each test network independently, serving as our single-network reference. **MOSAIC-zs** is pre-trained on the 64 networks in the data package, following the benchmark's $64/20$ split. To pre-train our model, we uniformly sample a token ID and feed the model a batch of graphs from that token with shuffled snapshots, i.e., without enforcing temporal order. We then evaluate MOSAIC-zs zero-shot on the 20 held-out test tokens and compare it against the baseline models.

Results. We first compare the models by ranking their test AUC on each of the 20 held-out networks, separately within the supervised and the zero-shot models, with rank 1 corresponding to the best performance. The resulting average rank measures how consistently a model performs relative to the other methods across networks. As reported in Table 5, MOSAIC-zs achieves the best average rank

Table 5: Ranking on the 20 MiNT test networks (3 seeds), computed separately for supervised and zero-shot models by test AUC (1 = best, 3 = worst). Per-network AUCs and transfer gains are reported in Appendix B.1.

Model	Avg. AUC ↑	Avg. Rank ↓	# 1st ↑	# 2nd ↑
GC-LSTM	0.609	2.10	7/20	4/20
HTGN	0.684	1.65	9/20	9/20
MOSAIC	0.618	2.25	4/20	7/20
GC-LSTM-zs	0.662	2.70	2/20	2/20
HTGN-zs	0.727	1.75	6/20	13/20
MOSAIC-zs (Ours)	**0.744**	**1.55**	**12/20**	5/20

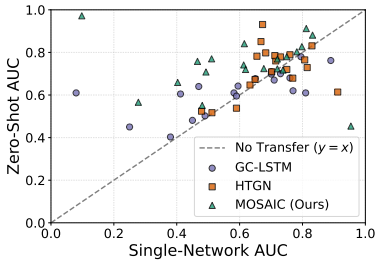

Figure 2: Per-network transfer on the 20 MiNT test networks: single-network AUC (x-axis) vs. zero-shot AUC (y-axis), with the diagonal indicating no transfer.

Table 6: Per-network transfer gain $\Delta_i = \mathrm{AUC}_i^{\mathrm{multi}} - \mathrm{AUC}_i^{\mathrm{single}}$ on the 20 MiNT test networks, with each pre-trained model compared against its own single-network backbone. $\Delta_i < 0$ indicates negative transfer.

Statistic	GC-LSTM-zs	HTGN-zs	MOSAIC-zs
Mean $\Delta \uparrow$	$+0.0536$	$+0.0424$	$\mathbf{+0.1258}$
Median $\Delta \uparrow$	$+0.0165$	$+0.0370$	$\mathbf{+0.0850}$
# networks with $\Delta < 0 \downarrow$	$6/20$	$4/20$	$\mathbf{2/20}$
# networks with $\Delta > +0.1 \uparrow$	$4/20$	$5/20$	$\mathbf{9/20}$

(1.65), ranking first on 12 networks and second on 5. We report in Appendix B.1 the per-network AUC. We also report in Appendix B.2 the comparison between supervised training of MOSAIC with the employed baselines.

The within-architecture comparison isolates the effect of multi-network pretraining. The improvement in average AUC is modest for HTGN (from 0.684 to 0.727), slightly larger for GC-LSTM (from 0.609 to 0.662), and largest for MOSAIC (from 0.618 to 0.744). Thus, this confirms that for this benchmark we benfit from our composition of priors and the adopted pre-training strategy.

Quantifying negative transfer. We next quantify these improvements through the per-network transfer gain $\Delta_i = \mathrm{AUC}_i^{\mathrm{multi}} - \mathrm{AUC}_i^{\mathrm{single}}$, where $\Delta_i < 0$ indicates negative transfer. Each model is compared against its own single-network counterpart. As Table 6 shows, MOSAIC-zs achieves a mean gain of $+0.126$, compared with $+0.042$ for HTGN-zs and $+0.054$ for GC-LSTM-zs. The gains are also more consistent: only $2/20$ networks show negative transfer for MOSAIC, compared with $4/20$ for HTGN and $6/20$ for GC-LSTM. Moreover, $9/20$ networks improve by more than 0.1 AUC with MOSAIC, compared with $5/20$ and $4/20$ for HTGN and GC-LSTM, respectively. Finally, Figure 2 visualizes these per-network gains by plotting zero-shot AUC against the corresponding single-network AUC. The diagonal represents no transfer, so points above it indicate positive transfer and points below it negative transfer. The concentration of MOSAIC points above the diagonal, together with several large vertical gains, provides a direct view of the magnitude and consistency of the improvements quantified in Table 6.

5 Conclusions

In this work, we introduced MOSAIC, a temporal graph learning framework that separates temporal representation extraction from trainable spatial modeling. By combining pretrained temporal representations, local temporal statistics, and supra-Laplacian positional encodings, MOSAIC avoids backpropagating through time and learns from these priors' patterns. Experiments show that our proposed composition provides complementary information, yielding competitive or state-of-the-art performance in supervised forecasting and robust zero-shot transfer across 20 unseen MiNT networks, outperforming all baselines. Ablation studies further show that the performance gains result from the combination of these priors rather than from any individual component. Future work can extend the framework to Continuous-Time Dynamic Graphs (C-TDGs) (Gravina et al., 2024), relax the channel-independence assumption during temporal feature extraction, and scale the pre-training corpus across heterogeneous domains.

AI use statement

In this work, we used generative AI tools for (i) improving the formatting and readability of the paper, (ii) implementing and editing software code for collecting experimental results, and (iii) assisting with the execution and scheduling of experiments. We did not use generative AI tools to derive the core methodology, design or provide feedback on the research methodology or experiments, generate or refine hypotheses, interpret experimental results, or generate references. All AI-assisted work

was reviewed by the authors: AI-assisted code was manually inspected and tested, and experimental results were independently verified. The authors reviewed and validated the final methodology, claims, results, conclusions, and presentation, and take full responsibility for the final content of this work, including all text, claims, code, results, and other artifacts produced with the aid of generative AI.

Ethics statement

This work does not involve human subjects or the collection of new personal or sensitive data. We use publicly available datasets introduced in previous research. We follow the usage conditions and licensing requirements associated with these datasets. The work does not introduce interventions involving individuals or applications intended to cause harm. We are not aware of any specific discrimination, fairness, privacy, security, or legal concerns arising from the use of these datasets or from the methodology presented in this work. The authors have no relevant conflicts of interest or sponsorship to disclose. We take responsibility for the integrity of the experimental methodology, data processing, and reported results.

Reproducibility statement

We will release the source code and experimental configurations upon acceptance. The paper and appendix provide detailed descriptions of the MOSAIC architecture, prior construction, training procedure, datasets, pre-processing steps, and experimental settings. We additionally report hyperparameter configurations, ablation and sensitivity analyses, and computational requirements to facilitate reproduction of the reported results.

References

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting. *ISPRS International Journal of Geo-Information*, 10(7), 2021. ISSN 2220-9964. doi: 10.3390/ijgi10070485.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *arXiv preprint arXiv:2007.02842*, 2020.

Yuanchen Bei, Weizhi Zhang, Siwen Wang, Weizhi Chen, Sheng Zhou, Hao Chen, Yong Li, Jiajun Bu, Shirui Pan, Yizhou Yu, Irwin King, Fakhri Karray, and Philip S. Yu. Graphs meet ai agents: Taxonomy, progress, and future opportunities. *arXiv preprint arXiv:2506.18019*, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Ayan Chatterjee, Barbara Ikica, Babak Ravandi, and John Palowitch. Transfer learning for temporal link prediction. *arXiv preprint arXiv:2504.10925*, 2025.

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: Mining loop detector data. *Transportation Research Record*, 1748(1): 96–102, 2001. doi: 10.3141/1748-12. URL https://doi.org/10.3141/1748-12.

Jinyin Chen, Xueke Wang, and Xuanheng Xu. GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction. *Applied Intelligence*, pp. 1–16, 2022.

Yuzhou Chen, Ignacio Segovia-Dominguez, and Yulia R. Gel. Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. *arXiv preprint arXiv:2105.04100*, 2021.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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)*, pp. 1724–1734. Association for Computational Linguistics, October 2014. doi: 10.3115/v1/D14-1179.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Andrea Cini, Ivan Marisca, Filippo Maria Bianchi, and Cesare Alippi. Scalable Spatiotemporal Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6): 7218–7226, Jun. 2023. doi: 10.1609/aaai.v37i6.25880.

Razvan-Gabriel Cirstea, Tung Kieu, Chenjuan Guo, Bin Yang, and Sinno Jialin Pan. Enhancenet: Plugin neural networks for enhancing correlated time series forecasting. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 1739–1750, 2021. doi: 10.1109/ICDE51399.2021.00153.

Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting–full version. *arXiv preprint arXiv:2203.15737*, 2022. URL https://arxiv.org/abs/2203.15737.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: fast and memory-efficient exact attention with io-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Fabrizio De Castelli, Alessio Gravina, Moshe Eliasof, Carola-Bibiane Schönlieb, and Davide Bacciu. Adaptive memory retention in dynamic graphs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=x7NGYPNlgs.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 269–278, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467330. URL https://doi.org/10.1145/3447548.3467330.

Vijay Prakash Dwivedi and Xavier Bresson. A Generalization of Transformer Networks to Graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Federico Errica, Alessio Gravina, Davide Bacciu, and Alessio Micheli. Hidden Markov Models for Temporal Graph Representation Learning. In *Proceedings of the 31st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2023.

Jin Fan, Wenchao Weng, Hao Tian, Huifeng Wu, Fu Zhu, and Jia Wu. Rgdan: A random graph diffusion attention network for traffic prediction. *Neural Networks*, 172:106093, 2024. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.106093.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ODE networks for traffic flow forecasting. *CoRR*, abs/2106.12931, 2021. URL https://arxiv.org/abs/2106.12931.

Yaniv Galron, Fabrizio Frasca, Haggai Maron, Eran Treister, and Moshe Eliasof. Understanding and improving laplacian positional encodings for temporal gnns. In Rita P. Ribeiro, Bernhard Pfahringer, Nathalie Japkowicz, Pedro Larrañaga, Alípio M. Jorge, Carlos Soares, Pedro H. Abreu, and João Gama (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track*, pp. 420–437, Cham, 2026a. Springer Nature Switzerland. ISBN 978-3-032-05981-9.

Yaniv Galron, Hadar Sinai, Haggai Maron, and Moshe Eliasof. Learning from historical activations in graph neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=8SnAGYf2wM.

Jianfei Gao and Bruno Ribeiro. On the equivalence between temporal and static equivariant graph representations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7052–7076. PMLR, 17–23 Jul 2022.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for Quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *ICML'17*, pp. 1263–1272. JMLR.org, 2017.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. Flowstate: Sampling-rate-equivariant time-series forecasting. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=9yNGBueKgZ.

Alessio Gravina and Davide Bacciu. Deep Learning for Dynamic Graphs: Models and Benchmarks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11788–11801, 2024. doi: 10.1109/TNNLS.2024.3379735.

Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. Long Range Propagation on Continuous-Time Dynamic Graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16206–16225. PMLR, 21–27 Jul 2024.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 855–864. Association for Computing Machinery, 2016. ISBN 9781450342322. doi: 10.1145/2939672.2939754.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):922–929, Jul. 2019. doi: 10.1609/aaai.v33i01.3301922.

Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2022. doi: 10.1109/TKDE.2021.3056502.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i4.25556. URL https://doi.org/10.1609/aaai.v37i4.25556.

Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*, 2024.

Yannis Karmim, Marc Lafon, Raphaël Fournier S' niehotta, and Nicolas Thome. Supra-laplacian encoding for transformer on dynamic graphs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 17215–17246. Curran Associates, Inc., 2024. doi: 10.52202/079017-0547.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation Learning for Dynamic Graphs: a Survey. *J. Mach. Learn. Res.*, 21 (1), jan 2020. ISSN 1532-4435.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 09 2014. ISSN 2051-1310. doi: 10.1093/comnet/cnu016. URL https://doi.org/10.1093/comnet/cnu016.

Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2): 517–541, 2001. doi: 10.1137/S1064827500366124. URL https://doi.org/10.1137/S1064827500366124.

Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martín-Martín. Modeling dynamic environments with scene graph memory. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17976–17993. PMLR, 23–29 Jul 2023.

Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11906–11917. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/lan22a.html.

Guohao Li, Matthias Müller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? *arXiv preprint arXiv:1904.03751*, 2019a.

Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. Predicting Path Failure In Time-Evolving Graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1279–1289, New York, NY, USA, 2019b. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330847.

Jintang Li, Zhouxin Yu, Zulun Zhu, Liang Chen, Qi Yu, Zibin Zheng, Sheng Tian, Ruofan Wu, and Changhua Meng. Scaling up dynamic graph representation learning via spiking neural networks. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i7.26034. URL https://doi.org/10.1609/aaai.v37i7.26034.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *Advances in Neural Information Processing Systems*, 37:127030–127058, 2024.

Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4189–4196, May 2021. doi: 10.1609/aaai.v35i5.16542. URL https://ojs.aaai.org/index.php/AAAI/article/view/16542.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. Open spatio-temporal foundation models for traffic prediction. *ACM Trans. Intell. Syst. Technol.*, 2025. ISSN 2157-6904. doi: 10.1145/3773912.

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*, 2024.

Antonio Longa, Veronica Lachi, Gabriele Santin, Monica Bianchini, Bruno Lepri, Pietro Lio, Franco Scarselli, and Andrea Passerini. Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. *arXiv preprint arXiv:2302.01018*, 2023.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S. Yu, and Yanfang Ye. Temporal network embedding with micro- and macro-dynamics. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pp. 469–478, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3357943. URL https://doi.org/10.1145/3357384.3357943.

Alessio Micheli and Domenico Tortorella. Discrete-time dynamic graph echo state networks. *Neurocomputing*, 496(C):85–95, jul 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2022.05.001.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Jiaxin Pan, Mojtaba Nayyeri, Osama Mohammed, Daniel Hernandez, Rongchuan Zhang, Cheng Cheng, and Steffen Staab. Towards foundation model on temporal knowledge graph reasoning. *arXiv preprint arXiv:2506.06367*, 2025.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 701–710. Association for Computing Machinery, 2014. ISBN 9781450329569. doi: 10.1145/2623330.2623732.

Markus N. Rabe and Charles Staats. Self-attention does not need $o(n^2)$ memory. *arXiv preprint arXiv:2112.05682*, 2022.

Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzmán López, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 4564–4573. Association for Computing Machinery, 2021. ISBN 9781450384469. doi: 10.1145/3459637.3482014.

Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing*, 68:6303–6318, 2020.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. ISSN 1476-4687. doi: 10.1038/323533a0.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *25th International Conference on Neural Information Processing*, ICONIP, pp. 362–373. Springer, 2018.

Kiarash Shamsi, Farimah Poursafaei, Shenyang Huang, Bao Tran Gia Ngo, Baris Coskunuzer, and Cuneyt Gurcan Akcora. Graphpulse: Topological representations for temporal graph property prediction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DZqic2sPTY.

Kiarash Shamsi, Tran Gia Bao Ngo, Razieh Shirzadkhani, Shenyang Huang, Farimah Poursafaei, Poupak Azad, Reihaneh Rabbany, Baris Coskunuzer, Guillaume Rabusseau, and Cuneyt Gurcan Akcora. MiNT: Multi-network transfer benchmark for temporal graph learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=Za7IcsXIRV.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=WEHSlH5mOk.

Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 1567–1577. ACM, 2022.

Uriel Singer, Ido Guy, and Kira Radinsky. Node embedding over temporal graphs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pp. 4605–4612. AAAI Press, 2019. ISBN 9780999241141.

Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. Space-time-separable graph convolutional network for pose forecasting, 2021.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):914–921, Apr. 2020. doi: 10.1609/aaai.v34i01.5438.

Aynaz Taheri and Tanya Berger-Wolf. Predictive temporal embedding of dynamic graphs. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '19, pp. 57–64. Association for Computing Machinery, 2020. ISBN 9781450368681. doi: 10.1145/3341161.3342872.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pp. 1907–1913. AAAI Press, 2019. ISBN 9780999241141.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 753–763. Association for Computing Machinery, 2020. ISBN 9781450379984. doi: 10.1145/3394486.3403118.

Dongkuan Xu, Wei Cheng, Dongsheng Luo, Xiao Liu, and Xiang Zhang. Spatio-temporal attentive rnn for node classification in temporal attributed graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 3947–3953. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/548. URL https://doi.org/10.24963/ijcai.2019/548.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/xu18c.html.

Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11592–11602. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xu21k.html.

Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 1975–1985, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467422. URL https://doi.org/10.1145/3447548.3467422.

Jiaxuan You, Tianyu Du, and Jure Leskovec. ROLAND: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2358–2366, 2022.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 3634–3640. AAAI Press, 2018. ISBN 9780999241127.

Yuan Yuan, Jingtao Ding, Chonghua Han, Zhi Sheng, Depeng Jin, and Yong Li. Uniflow: A foundation model for unified urban spatio-temporal flow prediction. *arXiv preprint arXiv:2411.12972*, 2025.

Xudong Zhang, Xuewen Chen, Haina Tang, Yulei Wu, Hanji Shen, and Jun Li. Adpstgcn: Adaptive spatial–temporal graph convolutional network for traffic forecasting. *Knowledge-Based Systems*, 301:112295, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2024.112295.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (01):1234–1241, Apr. 2020. doi: 10.1609/aaai.v34i01.5477.

Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Yifan Zhu, Fangpeng Cong, Dan Zhang, Wenwen Gong, Qika Lin, Wenzheng Feng, Yuxiao Dong, and Jie Tang. Wingnn: Dynamic graph neural networks with random gradient aggregation window. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 3650–3662. Association for Computing Machinery, 2023. ISBN 9798400701030. doi: 10.1145/3580305.3599551.

Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 2857–2866, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220054. URL https://doi.org/10.1145/3219819.3220054.

A Experimental Details

In this section, we provide details regarding our experiments.

A.1 Implementation Details

To ensure the full reproducibility of our framework, we detail the specific configurations and hyper-parameters used to instantiate MOSAIC across our experiments.

Time Series Foundation Model (Φ_{f}). For the extraction of temporal features, we use `Moment-1-small` from the family of open time series foundation models proposed in Goswami et al. (2024) as our temporal encoder Φ_{f}. The temporal embeddings were generated without fine-tuning the base model to fully exploit its pre-learned, domain-agnostic temporal priors. For statistical summaries we choose $\Phi_{\mathrm{s}} = \mathbf{I}$ for all spatio-temporal traffic forecasting benchmarks and we use adaptive average pooling to project the input window to 16 output dimensions for all other tasks.

Supra-Laplacian Hyperparameters. The multi-layer eigendecomposition requires setting the inter-layer temporal coupling weight μ and the number of lowest-frequency eigenmodes k. Across our evaluations, we set $\mu = 1$ to avoid unbalancing inter-layer information flow, and extracted the $k = 8$ smallest eigenpairs for all datasets. Since in spatio-temporal data the topology is not evolving, we perform random edge drops and insertions.

Optimization. Models were trained using the Adam optimizer with an initial learning rate of 10^{-3}, a weight decay of 10^{-4}, and a dropout probability of 0.1 shared among layers. We also apply cosine annealing (Loshchilov & Hutter, 2017) to the learning rate for a maximum of 200 or 300 epochs, depending on the task.

Data normalization. For spatio-temporal tasks we adopt a two-stage normalization procedure inspired by Reversible Instance Normalization (Kim et al., 2022). In particular, we first perform a global z-score normalization computing the mean $\mu^{(g)}$ and standard deviation $\sigma^{(g)}$ of each of the d_n channels independently, aggregating over the entire temporal and spatial dimensions of the training set. The raw inputs are transformed into globally normalized sequences $\tilde{\mathbf{X}} = (\mathbf{X} - \mu^{(g)})/\sigma^{(g)}$.

Then, at training time, we extract the local statistical context of each input window of length w_s. For a given node $v \in \mathcal{V}$ and its temporal window $[\tilde{\mathbf{X}}_{t-w_s+1:t}]_v$, we compute the local temporal mean $\mu_v^{(l)}$ and standard deviation $\sigma_v^{(l)}$ strictly along the time dimension. Real-world spatio-temporal datasets, such as those used for traffic forecasting, are prone to sensor failures that result in missing or constant flatline readings over a given window. Under these conditions, the local variance collapses to zero, which can cause numerical instability and severe distribution shifts during denormalization. To ensure robustness against such anomalies, we bypass scaling for flatlined sensors by replacing near-zero standard deviations with ones. Specifically, we introduce a threshold ϵ and apply the following fallback mechanism:

$$\hat{\sigma}_v^{(l)} = \begin{cases} \sigma_v^{(l)} & \text{if } \sigma_v^{(l)} > \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

The final predictions $\hat{\mathbf{Y}}$ in the original physical magnitude are reconstructed by directly applying the sequential reversal of both the local window statistics and the global dataset statistics to the model's raw predictions $\hat{\mathbf{Y}}_{\text{raw}}$:

$$\hat{\mathbf{Y}} = \underbrace{\left(\hat{\mathbf{Y}}_{\text{raw}} \odot \hat{\sigma}^{(l)} + \mu^{(l)}\right)}_{\text{Local Denormalization}} \odot \sigma^{(g)} + \mu^{(g)} \tag{9}$$

where $\odot$ denotes element-wise multiplication. This adaptive target scaling ensures the model inherently adjusts to localized temporal spikes while maintaining stability against anomalous, flat-lining sensor data.

Gate. To integrate the temporal positional encodings Φ_{p} into the joint latent space we employ an adaptive gating mechanism evaluated for each node. Let $\mathbf{z}_v \in \mathbb{R}^d$ represent the projected features of node $v \in \mathcal{V}$ in the joint latent space. First, the positional encoding vector for node v, denoted as $[\Phi_{\mathrm{p}}]_v$, is mapped to the hidden dimension d through a projection network, followed by a learnable

affine transformation with a strictly positive scale to ensure stability:

$$[\tilde{\Phi}_{\mathrm{p}}]_v = \mathrm{Softplus}(\mathbf{s}) \odot \mathrm{MLP}_{\mathrm{proj}}([\Phi_{\mathrm{p}}]_v) + \mathbf{b} \tag{10}$$

where $\mathbf{s}, \mathbf{b} \in \mathbb{R}^d$ are learnable scale and shift parameters shared across all nodes, $\odot$ denotes element-wise multiplication, and $\mathrm{MLP}_{\mathrm{proj}}$ is a shared projection module comprising layer normalization, a linear mapping, and a ReLU activation.

Next, we compute a node-specific, feature-conditioned gate that dynamically controls the amount of structural information injected into node v. The gate is derived from the concatenation of the node features and its processed positional encoding:

$$\mathbf{g}_v = \sigma\Big(\mathrm{Clamp}\big(\mathrm{MLP}_{\mathrm{gate}}([\mathbf{z}_v \,\|\, [\tilde{\Phi}_{\mathrm{p}}]_v]), -10, 10\big)\Big) \tag{11}$$

where $\|$ denotes concatenation along the feature dimension, $\mathrm{MLP}_{\mathrm{gate}}$ is a two-layer network with a GELU activation, Clamp bounds the pre-activation logits to $[-10, 10]$ to prevent numerical instability and gradient saturation, and σ is the sigmoid activation function.

Finally, the gated positional encodings are added to the original node features via a residual connection to produce the updated representation for node v:

$$\mathbf{z}'_v = \mathbf{z}_v + \mathbf{g}_v \odot [\tilde{\Phi}_{\mathrm{p}}]_v \tag{12}$$

This formulation ensures that the model can selectively attend to the structural and temporal priors provided by Φ_{p} based on the current context of each individual node's features, seamlessly fusing them into a final joint latent representation $\mathbf{z}'_v$ per node.

A.2 Employed Baselines

To evaluate the effectiveness of MOSAIC, we compare it against a comprehensive set of baselines. Specifically:

A3T-GCN (Bai et al., 2021), **AdpSTGCN** (Zhang et al., 2024), **AGCRN** (Bai et al., 2020), **ASTGCN** (Guo et al., 2019), **ASTGNN** (Guo et al., 2022), **DCRNN** (Li et al., 2018), **DeepWalk** (Perozzi et al., 2014), **DSTAGNN** (Lan et al., 2022), **DynamicTriad** (Zhou et al., 2018), **DynGESN** (Micheli & Tortorella, 2022), **DyGrAE** (Taheri & Berger-Wolf, 2020), **EnhanceNet** (Cirstea et al., 2021), **EvolveGCN-H** and **EvolveGCN-O** (Pareja et al., 2020), **GC-LSTM** (Chen et al., 2022), **GCRN-GRU** and **GCRN-LSTM** (Seo et al., 2018), **GGRNN** (Ruiz et al., 2020), **GMAN** (Zheng et al., 2020), **GraphPulse** (Shamsi et al., 2024), **GraphSSM** (Li et al., 2024), Graph WaveNet (**GWNet**) (Wu et al., 2019), **GTS** (Shang et al., 2021), **HMM4G** (Errica et al., 2023), **HTGN** (Yang et al., 2021), **HTNE** (Zuo et al., 2018), **M2DNE** (Lu et al., 2019), **MiNT** (Shamsi et al., 2026), **MPNN-LSTM** (Gilmer et al., 2017), **MTGNN** (Wu et al., 2020), **Node2Vec** (Grover & Leskovec, 2016), **PDFormer** Jiang et al. (2023), **Persistence Forecast** (Shamsi et al., 2026), **RGDAN** (Fan et al., 2024), **ROLAND** (You et al., 2022), **SpikeNet** (Li et al., 2023), **STAR** (Xu et al., 2019), **STEP** (Shao et al., 2022), **STFGNN** (Li & Zhu, 2021), **STGCN** (Yu et al., 2018), **STGODE** (Fang et al., 2021), **STNorm** (Deng et al., 2021), **STSGCN** (Sofianos et al., 2021), **ST-WA** (Cirstea et al., 2022), **Transformer** (Vaswani et al., 2017), **T-GCN** (Zhao et al., 2020), **tNodeEmbed** (Singer et al., 2019), **WinGNN** (Zhu et al., 2023), **Z-GCNETs** (Chen et al., 2021).

A.3 Employed Datasets

In our experiments, we evaluate MOSAIC on a variety of dynamic graph benchmarks. For node-level forecasting, we employ **Wikipedia Math**, which tracks daily page views; **ChickenPox Hungary**, representing epidemiological spread across counties; **Twitter Tennis**, modeling social interaction intensity; the large-scale traffic speed datasets **Metr-LA** and **PeMS-Bay**, which require capturing fine-grained correlations in sensor networks to predict the speed of passing vehicles; and the traffic flow benchmarks **PeMS03**, **PeMS04**, **PeMS07**, and **PeMS08**, which measure vehicle counts across California highway sensor networks at different geographic scales. For node classification on discrete-time dynamic graphs we additionally employ **DBLP-10** and **arXiv**, two citation networks in which snapshots differ in topology while sharing node attributes. We follow the transductive protocol of Li et al. (2024): an $80\%/20\%$ stratified node split on DBLP-10 and the fixed public splits on arXiv,

Table 7: Employed datasets statistics

Dataset	#Graphs	#Nodes	#Edges	Sequence Length	Evolving Topology
Twitter Tennis	1	1000	40839	120	✓
ChickenPox	1	20	102	517	✗
Wikipedia Math	1	1068	27079	723	✗
Metr-LA	1	207	1515	34272	✗
PeMS-BAY	1	325	2369	52116	✗
PeMS03	1	358	1093	26208	✗
PeMS04	1	307	680	16992	✗
PeMS07	1	883	1732	28224	✗
PeMS08	1	170	548	17856	✗
DBLP-10	1	28085	236894	27	✓
arXiv	1	169343	2315598	35	✓

evaluated with Micro- and Macro-F1 over 5 runs. In Table 7, we report the statistics of the employed datasets.

Instead, the statistics of the 84 employed token networks are shown in Table 8.

Table 8: All token networks' statistics (Shamsi et al., 2026).

Token	#Node	#Transaction	#Snapshots (days)	Growth rate	Novelty	Surprise	Token	#Node	#Transaction	#Snapshots (days)	Growth rate	Novelty	Surprise
ARC	11325	70968	606	0.43	0.32	0.88	Metis	52586	343141	907	0.44	0.48	0.89
CELR	65350	235807	1691	0.49	0.56	0.96	cDAI	52753	358050	1437	0.45	0.46	0.9
CMT	86895	205961	309	0.45	0.72	0.92	BITCOIN	34051	347054	178	0.48	0.39	0.63
DRGN	113453	341849	2164	0.44	0.57	0.97	INJ	60472	312822	1113	0.46	0.52	0.98
GHST	35156	180955	1146	0.43	0.51	0.93	MIM	23038	269366	885	0.44	0.4	0.89
INU	8556	66315	154	0.27	0.41	0.59	GLM	53385	234912	1080	0.5	0.53	0.96
IOTX	63079	288469	1993	0.45	0.56	0.99	Mog	14590	240680	107	0.37	0.38	0.55
QSP	117977	299671	2178	0.45	0.67	0.99	DPI	40627	234246	1150	0.49	0.5	0.86
REP	83282	224843	346	0.46	0.69	0.96	LINA	45342	227147	1144	0.45	0.46	0.95
RFD	23208	173695	169	0.3	0.39	0.6	Yf-DAI	22466	226875	1158	0.42	0.31	0.87
TNT	88247	316352	1216	0.43	0.55	0.93	BOB	42806	212099	199	0.35	0.48	0.73
TRAC	71667	299181	2110	0.46	0.54	0.97	RGT	35277	211932	1110	0.44	0.46	0.98
RLB	28033	240291	129	0.43	0.49	0.76	TVK	42539	208082	1062	0.41	0.48	0.93
steCRV	19079	211538	1033	0.45	0.53	0.9	RSR	50645	205906	659	0.47	0.62	0.91
ALBT	63042	434881	1152	0.43	0.44	0.89	WOJAK	34341	198653	201	0.37	0.48	0.73
POLS	128159	554705	1132	0.45	0.61	0.94	ANT	36517	200262	1107	0.47	0.46	0.93
SWAP	69230	509769	1213	0.46	0.45	0.79	LADYS	37486	192176	181	0.37	0.52	0.79
SUPER	83299	502030	986	0.47	0.46	0.85	ETH2x-FLI	11008	199088	965	0.47	0.28	0.84
RARI	87186	502960	1207	0.43	0.47	0.91	TURBO	38638	189048	189	0.33	0.48	0.72
KP3R	39323	493258	1102	0.43	0.33	0.88	REPv2	39061	191367	1194	0.48	0.5	0.97
MIR	79984	444998	1066	0.45	0.43	0.92	NOIA	29798	185528	1133	0.46	0.37	0.7
aUSDC	23742	475680	1067	0.46	0.4	0.73	0x0	21531	182430	283	0.51	0.46	0.81
LUSD	25852	430473	943	0.48	0.36	0.87	PSYOP	25450	168896	169	0.32	0.39	0.59
PICKLE	28498	430262	1149	0.48	0.34	0.69	ShibDoge	40023	134697	680	0.43	0.53	0.8
DODO	47046	390443	1131	0.47	0.45	0.91	ADX	14567	123755	1188	0.44	0.4	0.91
YFII	43964	391984	1196	0.44	0.44	0.96	BAG	11860	122634	298	0.31	0.44	0.87
STARL	71590	369913	856	0.46	0.48	0.86	QOM	21757	118292	598	0.46	0.41	0.81
LQTY	34687	374230	943	0.45	0.34	0.91	BEPRO	26521	120261	1132	0.46	0.48	0.87
FEG	118294	367584	1007	0.4	0.62	0.92	AIOZ	29231	119926	947	0.43	0.49	0.89
AUDIO	91218	362685	1108	0.45	0.58	0.95	PRE	40476	118625	1113	0.5	0.55	0.86
OHM	45728	377068	690	0.43	0.46	0.88	CRU	19990	117712	1144	0.5	0.43	0.95
WOOL	16874	351178	716	0.41	0.18	0.41	POOH	27245	111641	193	0.26	0.49	0.69
DERC	24277	111205	824	0.45	0.49	0.83	aDAI	13648	187050	1068	0.45	0.46	0.82
stkAAVE	37355	110924	1128	0.42	0.57	0.71	ORN	44010	239451	1134	0.46	0.47	0.87
BTRFLY	8450	108371	453	0.48	0.34	0.44	DOGE2.0	7664	79047	123	0.45	0.38	0.66
SDEX	9127	104869	240	0.41	0.44	0.75	HOICHI	5075	77361	436	0.36	0.32	0.71
XCN	20085	104185	607	0.46	0.42	0.84	EVERMOON	7552	79868	163	0.24	0.35	0.52
HOP	37004	102650	514	0.41	0.6	0.88	MUTE	12426	82345	977	0.43	0.46	0.95
MAHA	18401	96180	749	0.43	0.47	0.91	crvUSD	2950	88647	174	0.61	0.37	0.73
DINO	15837	94140	358	0.44	0.44	0.74	SLP	6675	95368	1151	0.43	0.36	0.91
bendWETH	1454	96898	593	0.51	0.21	0.51	sILV2	12838	92905	611	0.4	0.34	0.48
PUSH	14501	93103	936	0.46	0.38	0.83	SPONGE	25852	90468	184	0.31	0.66	0.81

A.4 Hyperparameter Space

The hyperparameter space employed by MOSAIC across all tasks is reported in Table 9.

B Additional Results and Comparisons

In this section we report additional results and comparisons with other datasets and baselines.

Table 9: The grid of hyperparameters employed by MOSAIC during model selection for all evaluated tasks. Datasets include: Traffic Forecasting (**PeMS03**, **PeMS04**, **PeMS07**, **PeMS08**), Node Prediction (**ChickenPox**, **Wikipedia Math**, **Twitter Tennis**), and the **MiNT** benchmark.

Hyperparameters	**Values**		
	Traffic Forecasting	**Node Prediction**	**MiNT**
Optimizer	Adam	Adam	Adam
Learning rate	0.003, 0.001	0.001	0.001
Weight decay	10^{-4}	10^{-4}	10^{-4}
d	512	256, 512	32, 64
L	10	5, 10	2
Aggregate layers	✓	✓	✗
w_f	12, 288	16, 32	8
w_s	$w_s = w_f$	$w_s = w_f$	$w_s = w_f$
w_p	8	8	8
k	8	8	8
MHSA #heads	2, 4	2, 4	2, 4

Table 10: Test AUC (↑) on the 20 held-out MiNT test networks, averaged over 3 seeds. Supervised models are trained on each test network and reported for reference only; zero-shot models are pre-trained on the 64 MiNT training networks and never see the test networks. Baseline results are reported from (Shamsi et al., 2026). **First** and **second** best results are evaluated within the zero-shot models only.

	Zero-Shot			**Supervised**		
Network	**MOSAIC-zs (Ours)**	HTGN-zs	GC-LSTM-zs	MOSAIC	HTGN	GC-LSTM
MIR	**$0.769_{\pm 0.049}$**	**$0.836_{\pm 0.016}$**	$0.788_{\pm 0.022}$	$0.720_{\pm 0.027}$	$0.750_{\pm 0.005}$	$0.768_{\pm 0.026}$
DOGE2.0	**$0.551_{\pm 0.022}$**	$0.538_{\pm 0.038}$	**$0.551_{\pm 0.022}$**	$0.481_{\pm 0.051}$	$0.590_{\pm 0.059}$	$0.538_{\pm 0.000}$
MUTE	**$0.741_{\pm 0.014}$**	**$0.673_{\pm 0.013}$**	$0.636_{\pm 0.003}$	$0.613_{\pm 0.107}$	$0.649_{\pm 0.015}$	$0.593_{\pm 0.030}$
EVERMOON	**$0.770_{\pm 0.025}$**	**$0.517_{\pm 0.039}$**	$0.494_{\pm 0.047}$	$0.511_{\pm 0.141}$	$0.512_{\pm 0.023}$	$0.562_{\pm 0.179}$
DERC	**$0.841_{\pm 0.028}$**	**$0.798_{\pm 0.027}$**	$0.696_{\pm 0.011}$	$0.615_{\pm 0.104}$	$0.683_{\pm 0.013}$	$0.703_{\pm 0.022}$
ADX	**$0.724_{\pm 0.005}$**	**$0.679_{\pm 0.024}$**	$0.673_{\pm 0.022}$	$0.677_{\pm 0.018}$	$0.769_{\pm 0.018}$	$0.723_{\pm 0.002}$
HOICHI	$0.723_{\pm 0.005}$	**$0.765_{\pm 0.018}$**	**$0.847_{\pm 0.005}$**	$0.720_{\pm 0.002}$	$0.807_{\pm 0.047}$	$0.857_{\pm 0.000}$
SDEX	$0.454_{\pm 0.014}$	**$0.614_{\pm 0.020}$**	**$0.723_{\pm 0.002}$**	$0.954_{\pm 0.032}$	$0.762_{\pm 0.034}$	$0.720_{\pm 0.002}$
BAG	**$0.972_{\pm 0.013}$**	**$0.931_{\pm 0.028}$**	$0.529_{\pm 0.023}$	$0.098_{\pm 0.036}$	$0.673_{\pm 0.227}$	$0.196_{\pm 0.179}$
XCN	**$0.758_{\pm 0.176}$**	**$0.851_{\pm 0.043}$**	$0.733_{\pm 0.003}$	$0.466_{\pm 0.074}$	$0.668_{\pm 0.099}$	$0.306_{\pm 0.092}$
ETH2x-FLI	**$0.881_{\pm 0.004}$**	**$0.729_{\pm 0.015}$**	$0.697_{\pm 0.009}$	$0.832_{\pm 0.033}$	$0.610_{\pm 0.059}$	$0.670_{\pm 0.009}$
stkAAVE	**$0.708_{\pm 0.023}$**	**$0.709_{\pm 0.022}$**	$0.650_{\pm 0.028}$	$0.493_{\pm 0.104}$	$0.702_{\pm 0.042}$	$0.368_{\pm 0.011}$
GLM	**$0.912_{\pm 0.006}$**	**$0.831_{\pm 0.024}$**	$0.501_{\pm 0.027}$	$0.812_{\pm 0.149}$	$0.830_{\pm 0.029}$	$0.451_{\pm 0.003}$
QOM	**$0.720_{\pm 0.025}$**	**$0.647_{\pm 0.019}$**	$0.618_{\pm 0.004}$	$0.619_{\pm 0.026}$	$0.633_{\pm 0.017}$	$0.612_{\pm 0.001}$
WOJAK	**$0.566_{\pm 0.006}$**	$0.524_{\pm 0.027}$	**$0.534_{\pm 0.020}$**	$0.278_{\pm 0.179}$	$0.479_{\pm 0.005}$	$0.484_{\pm 0.000}$
DINO	**$0.659_{\pm 0.048}$**	**$0.779_{\pm 0.113}$**	$0.659_{\pm 0.039}$	$0.403_{\pm 0.021}$	$0.730_{\pm 0.195}$	$0.874_{\pm 0.028}$
Metis	**$0.803_{\pm 0.008}$**	**$0.760_{\pm 0.025}$**	$0.697_{\pm 0.013}$	$0.782_{\pm 0.013}$	$0.715_{\pm 0.122}$	$0.646_{\pm 0.023}$
REPv2	**$0.780_{\pm 0.018}$**	**$0.789_{\pm 0.020}$**	$0.733_{\pm 0.019}$	$0.750_{\pm 0.006}$	$0.760_{\pm 0.012}$	$0.725_{\pm 0.014}$
TRAC	**$0.717_{\pm 0.010}$**	**$0.785_{\pm 0.008}$**	$0.742_{\pm 0.004}$	$0.738_{\pm 0.004}$	$0.712_{\pm 0.071}$	$0.748_{\pm 0.000}$
BEPRO	**$0.827_{\pm 0.034}$**	**$0.782_{\pm 0.003}$**	$0.746_{\pm 0.015}$	$0.798_{\pm 0.037}$	$0.655_{\pm 0.038}$	$0.632_{\pm 0.019}$
Mean AUC ↑	**0.744**	0.727	0.662	0.618	**0.684**	0.609
Avg. Rank ↓	**1.55**	1.75	2.70	2.25	**1.65**	2.10

B.1 Per-Network Zero-Shot Transfer on MiNT

Table 10 presents the detailed per-network zero-shot AUC evaluation across all 20 test networks in MiNT. Overall, MOSAIC-zs demonstrates dominant transferability, securing top performance on 11 out of 20 networks (ranking first on 7 networks and second on 4). Crucially, MOSAIC-zs outperforms HTGN-zs on 12 of the 20 target networks and shows better positive transfer. GC-LSTM-zs, instead, is less strong at learning transferable patterns from the pre-training corpus.

B.2 Supervised Evaluation on MiNT Networks

Table 11 evaluates the per-network supervised performance of MOSAIC trained from scratch on individual target networks against seven single-model baselines across the 20 MiNT test domains.

When trained in a purely supervised setup, MOSAIC demonstrates competitive capabilities, securing best performance on 3 networks (SDEX, ETH2x-FLI, BEPRO) and second best performance on 2 others (GLM and Metis). On SDEX and ETH2x-FLI, MOSAIC outperforms the best baseline by substantial margins of $+0.192$ and $+0.144$ AUC, respectively. Among the baselines, GraphPulse emerges as the strongest single-network model, achieving top rank on 10 out of 20 networks, while GC-LSTM (4 top ranks) and HTGN (2 top ranks) regularly finish among the top three. Conversely, WinGNN and the naive Persistence Forecast (P.F.) struggle across the suite, with P.F. yielding the lowest overall performance.

These results demonstrate that learning representations that are transferable across networks requires enough inputs. The massive positive transfer achieved through multi-network pre-training illustrates how single-network performance on BAG (0.098 AUC, i.e., inverted predictions) is brought to 0.972 AUC zero-shot.

Table 11: Supervised Evaluation Performance (AUC) on the 20 MiNT Test Networks. This table compares our fully trained MOSAIC against the single supervised baselines. Best results in **green**, second best in **blue**, and third best in **bold**.

	Single Model on Individual Networks							
Network	HTGN	GC-LSTM	EvolveGCN	GraphPulse	ROLAND	WinGNN	P.F.	**MOSAIC**
WOJAK	$0.479_{\pm 0.005}$	$0.484_{\pm 0.000}$	$\mathbf{0.505_{\pm 0.023}}$	$0.467_{\pm 0.030}$	$\mathbf{0.529}_{\pm 0.005}$	$\mathbf{0.511}_{\pm 0.026}$	0.378	$0.278_{\pm 0.179}$
DOGE2.0	$\mathbf{0.590}_{\pm 0.059}$	$0.538_{\pm 0.000}$	$\mathbf{0.551_{\pm 0.022}}$	$0.384_{\pm 0.180}$	$0.513_{\pm 0.022}$	$\mathbf{0.577}_{\pm 0.038}$	0.250	$0.481_{\pm 0.051}$
EVERMOON	$0.512_{\pm 0.023}$	$\mathbf{0.562}_{\pm 0.179}$	$0.451_{\pm 0.046}$	$\mathbf{0.519_{\pm 0.130}}$	$0.349_{\pm 0.119}$	$\mathbf{0.525}_{\pm 0.114}$	0.241	$0.511_{\pm 0.141}$
QOM	$0.633_{\pm 0.017}$	$0.612_{\pm 0.001}$	$0.618_{\pm 0.002}$	$\mathbf{0.775}_{\pm 0.011}$	$\mathbf{0.641_{\pm 0.003}}$	$\mathbf{0.645}_{\pm 0.099}$	0.334	$0.619_{\pm 0.026}$
SDEX	$\mathbf{0.762}_{\pm 0.034}$	$0.720_{\pm 0.002}$	$\mathbf{0.733_{\pm 0.028}}$	$0.436_{\pm 0.030}$	$0.483_{\pm 0.254}$	$0.726_{\pm 0.000}$	0.423	$\mathbf{0.954}_{\pm 0.032}$
ETH2x-FLI	$0.610_{\pm 0.059}$	$\mathbf{0.670_{\pm 0.009}}$	$\mathbf{0.688}_{\pm 0.010}$	$0.666_{\pm 0.047}$	$0.621_{\pm 0.023}$	$0.617_{\pm 0.056}$	0.355	$\mathbf{0.832}_{\pm 0.033}$
BEPRO	$0.655_{\pm 0.038}$	$0.632_{\pm 0.019}$	$0.610_{\pm 0.012}$	$\mathbf{0.783}_{\pm 0.003}$	$0.439_{\pm 0.125}$	$\mathbf{0.736_{\pm 0.018}}$	0.393	$\mathbf{0.798}_{\pm 0.037}$
XCN	$\mathbf{0.668_{\pm 0.099}}$	$0.306_{\pm 0.092}$	$0.512_{\pm 0.067}$	$\mathbf{0.821}_{\pm 0.004}$	$\mathbf{0.765}_{\pm 0.015}$	$0.586_{\pm 0.029}$	0.592	$0.466_{\pm 0.074}$
BAG	$\mathbf{0.673_{\pm 0.227}}$	$0.196_{\pm 0.179}$	$0.329_{\pm 0.040}$	$\mathbf{0.934}_{\pm 0.020}$	$0.418_{\pm 0.016}$	$0.485_{\pm 0.105}$	**0.792**	$0.098_{\pm 0.036}$
TRAC	$0.712_{\pm 0.071}$	$\mathbf{0.748_{\pm 0.000}}$	$\mathbf{0.748_{\pm 0.000}}$	$\mathbf{0.767}_{\pm 0.001}$	$0.495_{\pm 0.223}$	$\mathbf{0.752}_{\pm 0.007}$	0.400	$0.738_{\pm 0.004}$
DERC	$\mathbf{0.683_{\pm 0.013}}$	$\mathbf{0.703}_{\pm 0.022}$	$0.669_{\pm 0.009}$	$\mathbf{0.769}_{\pm 0.040}$	$0.405_{\pm 0.357}$	$0.674_{\pm 0.044}$	0.353	$0.615_{\pm 0.104}$
Metis	$\mathbf{0.715_{\pm 0.122}}$	$0.646_{\pm 0.023}$	$0.688_{\pm 0.027}$	$\mathbf{0.812}_{\pm 0.011}$	$0.696_{\pm 0.108}$	$0.690_{\pm 0.039}$	0.423	$\mathbf{0.782}_{\pm 0.013}$
REPv2	$\mathbf{0.760}_{\pm 0.012}$	$0.725_{\pm 0.014}$	$0.709_{\pm 0.002}$	$\mathbf{0.830}_{\pm 0.001}$	$\mathbf{0.751_{\pm 0.003}}$	$0.744_{\pm 0.026}$	0.321	$0.750_{\pm 0.006}$
DINO	$0.730_{\pm 0.195}$	$\mathbf{0.874}_{\pm 0.028}$	$\mathbf{0.868}_{\pm 0.029}$	$\mathbf{0.801_{\pm 0.020}}$	$0.497_{\pm 0.092}$	$0.628_{\pm 0.251}$	0.431	$0.403_{\pm 0.021}$
HOICHI	$0.807_{\pm 0.047}$	$\mathbf{0.857}_{\pm 0.000}$	$\mathbf{0.856}_{\pm 0.001}$	$0.714_{\pm 0.010}$	$\mathbf{0.815_{\pm 0.036}}$	$0.769_{\pm 0.101}$	0.374	$0.720_{\pm 0.002}$
MUTE	$\mathbf{0.649}_{\pm 0.015}$	$0.593_{\pm 0.030}$	$\mathbf{0.617_{\pm 0.010}}$	$\mathbf{0.779}_{\pm 0.004}$	$0.289_{\pm 0.042}$	$0.593_{\pm 0.054}$	0.536	$0.613_{\pm 0.107}$
GLM	$\mathbf{0.830}_{\pm 0.029}$	$0.451_{\pm 0.003}$	$0.501_{\pm 0.033}$	$\mathbf{0.769_{\pm 0.018}}$	$0.559_{\pm 0.357}$	$0.530_{\pm 0.004}$	0.427	$\mathbf{0.812}_{\pm 0.149}$
MIR	$\mathbf{0.750}_{\pm 0.005}$	$\mathbf{0.768}_{\pm 0.026}$	$\mathbf{0.745_{\pm 0.015}}$	$0.689_{\pm 0.097}$	$0.228_{\pm 0.060}$	$0.742_{\pm 0.015}$	0.327	$0.720_{\pm 0.027}$
stkAAVE	$\mathbf{0.702}_{\pm 0.042}$	$0.368_{\pm 0.011}$	$0.397_{\pm 0.022}$	$\mathbf{0.743}_{\pm 0.006}$	$\mathbf{0.591_{\pm 0.122}}$	$0.572_{\pm 0.018}$	0.426	$0.493_{\pm 0.104}$
ADX	$\mathbf{0.769}_{\pm 0.018}$	$0.723_{\pm 0.002}$	$0.718_{\pm 0.004}$	$\mathbf{0.784}_{\pm 0.002}$	$\mathbf{0.761_{\pm 0.011}}$	$0.733_{\pm 0.023}$	0.362	$0.677_{\pm 0.018}$
Top rank ↑	2	4	0	10	1	0	0	3

B.3 Traffic Speed Forecasting Benchmarks

In Table 12 we report the MaskedMAE, MaskedRMSE and MaskedMAPE of MOSAIC on the Metr-LA and PeMS-Bay benchmarks. Overall, MOSAIC demonstrates competitive MaskedRMSE, which penalizes large prediction errors. This specific result highlights the model's reliability in handling extreme variance and anomalous traffic conditions. While MOSAIC does not strictly dominate the top-performing models across all remaining MaskedMAE and MaskedMAPE metrics, this reflects a deliberate structural trade-off. MOSAIC operates with strong inductive biases, providing a highly reliable and theoretically grounded alternative.

Table 12: Time series forecasting errors (MAE, RMSE, MAPE) on the Metr-LA and PeMS-Bay datasets, masked and computed at step 12. **First**, **second**, and **third** best results are color-coded.

Model	Metr-LA			PeMS-Bay		
	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓
AdpSTGCN	**3.40**	**7.21**	**9.45%**	**1.92**	4.49	4.62%
ASTGCN	6.51	12.52	11.64%	2.61	5.42	6.00%
DCRNN	3.60	7.60	10.50%	2.07	4.74	4.90%
GMAN	3.44	7.35	10.07%	1.86	**4.32**	**4.37%**
Graph WaveNet	3.53	7.37	10.01%	1.95	4.52	4.63%
GTS	3.46	7.31	9.98%	1.95	**4.43**	4.58%
MTGNN	3.49	7.23	**9.87%**	1.94	4.49	4.53%
RGDAN	**3.26**	**7.02**	9.73%	**1.82**	**4.20**	**4.28%**
STEP	**3.37**	**6.99**	**9.61%**	**1.79**	**4.20**	**4.18%**
STGCN	4.59	9.40	12.70%	2.49	5.69	5.79%
STSGCN	5.06	11.66	12.91%	2.26	5.21	5.40%
GGRNN	3.88	8.14	10.59%	2.34	5.14	5.21%
GraphSSM-S4	3.74	7.90	10.37%	1.98	4.45	4.77%
MOSAIC	3.85	**7.21**	11.17%	2.10	**4.20**	5.02%

B.4 Sensitivity of the Supra-Laplacian Positional Encoding

The supra-Laplacian positional encodings we adopted as input prior have been shown to be crucial for achieving good generalization capabilities both in the supervised and zero-shot setting. Given that the encodings are dependent on three main parameters, which are the number of retained eigenvalues and eigenvectors k, the inter-layer connection sensitivity μ and the input context window w_p, we show that the performances we achieve are not tied to the specific choice of these. Table 13 reports Φ_p's sensitivity to (k, w_p, μ) on Twitter Tennis, Wikipedia Math, and PeMS04, where the first row represents the default configuration used in all experiments. We observe no substantial variation across combinations, confirming that the structural prior is robust to this choice.

Table 13: Sensitivity of Φ_p to (k, w_p, μ) on Twitter Tennis and Wikipedia Math (MSE ↓) and PeMS04 (MAE ↓). The first row is the configuration used in the main experiments.

k	w_p	μ	Twitter Tennis	Wiki Math	PeMS04
8	8	1	0.262 ± 0.002	0.440 ± 0.002	19.08
4	4	0.5	0.264 ± 0.004	0.452 ± 0.003	19.28
4	4	2	0.262 ± 0.002	0.445 ± 0.004	19.28
4	16	0.5	0.276 ± 0.008	0.451 ± 0.004	19.22
16	4	0.5	0.267 ± 0.001	0.443 ± 0.004	19.23
16	16	2	0.263 ± 0.003	0.444 ± 0.005	19.20

B.5 Instantiation of the Temporal Prior

The temporal prior Φ_f relies on a pretrained time-series foundation model, raising the question of whether its contribution is sensitive to the choice of pretrained backbone. To evaluate this, Table 14 replaces MOMENT with two alternative temporal foundation models while keeping all other components and hyperparameters fixed: Moirai (Woo et al., 2024) and the SSM-based foundation model FlowState (Graf et al., 2026). Replacing MOMENT with either backbone results in essentially unchanged performance across all seven datasets. This demonstrates that the effectiveness of Φ_f is not specific to a particular foundation model, but is instead consistent across different pretrained temporal backbones. Together with the ablation study in Section 4.2, which shows that removing the foundation model substantially degrades performance, these results indicate that our approach benefits from the use of pretrained temporal representations while remaining robust to the choice of their underlying architecture.

Table 14: Sensitivity of MOSAIC to the instantiation of the temporal prior Φ_f on the supervised forecasting benchmarks. MSE (↓) for ChickenPox, Twitter Tennis, and Wiki Math; MAE (↓) for PeMS03/04/07/08. Every other component and hyperparameter is fixed at its optimal value; only Φ_f is swapped, and each alternative is kept frozen.

Φ_f	**ChickenPox** MSE ↓	**Tennis** MSE ↓	**Wiki Math** MSE ↓	**PeMS03** MAE ↓	**PeMS04** MAE ↓	**PeMS07** MAE ↓	**PeMS08** MAE ↓
MOMENT (default)	0.745	0.262	0.440	14.68	19.08	19.68	14.09
Moirai (Woo et al., 2024)	0.862	0.263	0.446	14.76	19.25	19.85	14.38
FlowState (Graf et al., 2026)	0.810	0.278	0.456	14.49	19.06	19.44	13.73

B.6 Runtime and Memory Analysis

The factorization of temporal processing into a gradient-free preprocessing stage raises the question of whether the resulting training efficiency comes at the cost of a substantial preprocessing overhead or increased memory requirements. We therefore separately quantify the cost of the supra-Laplacian positional encoding and the training-time computational and memory requirements of MOSAIC. These measurements complement the complexity analysis in the main paper by characterizing the practical cost of each phase.

B.6.1 Pre-processing

Table 15 reports the one-time cost of the supra-Laplacian positional encoding on three MiNT token networks spanning two orders of magnitude in size. The cost per snapshot grows linearly with $\mathrm{nnz}(\mathbf{L}_{\mathrm{sup}})$, matching the $\mathcal{O}(k\,\mathrm{nnz}(\mathbf{L}_{\mathrm{sup}}))$ behavior of LOBPCG (Knyazev, 2001) at fixed k, while its memory footprint is $\mathcal{O}(\mathrm{nnz}(\mathbf{L}_{\mathrm{sup}}) + knw_p)$. The largest network in the benchmark is encoded in under 20 minutes on a single CPU core within 1.1 GB of resident memory. Since the stage is gradient-free and independent across temporal windows, this cost is paid once per dataset and parallelizes directly.

Table 15: One-time cost of the supra-Laplacian positional encoding on three MiNT token networks of increasing size, measured on a single CPU core with $k = 8$ retained eigenpairs and window length $w_p = 8$. Peak RSS denotes the peak resident memory of the pre-computation stage.

Token	n	**Snapshots**	$\mathrm{nnz}(\mathbf{L}_{\mathrm{supra}})$	**s / snapshot**	**Total**	**Peak RSS**
POLS	128,159	1,132	1,856,788	1.02	18.6 min	1.09 GB
GLM	53,385	1,080	759,946	0.34	6.1 min	0.80 GB
HOICHI	5,075	436	78,423	0.12	0.8 min	0.64 GB

B.6.2 Training time and allocated memory

Table 16 compares seconds per epoch and peak GPU memory against DCRNN, GraphWaveNet, AGCRN, and EvolveGCN on PeMS04, using batches of 128 graphs of 307 nodes and a forecasting horizon of 12. MOSAIC is the fastest configuration at every input window W, $6.9\times$ faster than DCRNN at $W{=}12$ and $31\times$ faster at $W{=}48$. Its training cost stays flat in W because the temporal priors are precomputed and no gradient flows through time, whereas every baseline scales with the window it must backpropagate through. The same asymmetry appears in memory: MOSAIC grows 0.08 GB per additional input step against 0.75 GB for DCRNN and 0.37 GB for AGCRN, both of which exhaust the device at $W{=}96$.

Table 16: Training cost (seconds/epoch) and peak GPU memory (CUDA GB, in brackets) on PeMS04 as a function of the input window W, using batches of 128 graphs of 307 nodes and a forecasting horizon of 12. MOSAIC is fastest at every window, and its cost is flat in W since no gradient flows through the precomputed temporal priors. OOM denotes out-of-memory on the evaluation device.

W	**MOSAIC**	**DCRNN**	**GraphWaveNet**	**AGCRN**	**EvolveGCN**
12	2.0 [2.3]	13.8 [9.1]	10.9 [4.8]	6.1 [4.5]	2.8 [2.2]
24	2.0 [3.3]	29.0 [18.1]	23.2 [12.6]	13.2 [8.9]	6.6 [4.5]
48	2.0 [5.2]	63.1 [36.0]	52.5 [29.4]	30.8 [17.8]	17.5 [9.0]
96	2.1 [9.0]	OOM	OOM	79.5 [35.8]	52.5 [18.1]

C Proof of the supra-Laplacian variational formulation

We derive the variational formulation of the supra-Laplacian positional encoding by expanding its quadratic form over the temporal layers. Let $\mathbf{U} = [\mathbf{U}_{t-w_p+1}^\top, \dots, \mathbf{U}_t^\top]^\top$ denote the matrix containing the k eigenvectors of $\mathbf{L}_{\text{sup}}$, and introduce the change of variables

$$\mathbf{V} = \mathbf{D}_{\text{sup}}^{-\frac{1}{2}}\mathbf{U}, \qquad \mathbf{V} = \left[\mathbf{V}_{t-w_p+1}^\top, \dots, \mathbf{V}_t^\top\right]^\top. \tag{13}$$

This transformation is useful because it converts the normalized supra-Laplacian quadratic form into the quadratic form of the combinatorial supra-Laplacian. In particular,

$$\mathbf{U}^\top\mathbf{U} = \mathbf{I} \quad \Longleftrightarrow \quad \mathbf{V}^\top\mathbf{D}_{\text{sup}}\mathbf{V} = \mathbf{I}, \tag{14}$$

and

$$\mathbf{U}^\top\mathbf{L}_{\text{sup}}\mathbf{U} = \mathbf{V}^\top\left(\mathbf{D}_{\text{sup}} - \mathbf{A}_{\text{sup}}\right)\mathbf{V}. \tag{15}$$

We next write the supra-adjacency and supra-degree matrices explicitly. Because consecutive layers are connected one-to-one with uniform strength μ, the supra-adjacency is block tridiagonal,

$$\mathbf{A}_{\text{sup}} = \begin{pmatrix} \mathbf{A}_{t-w_p+1} & \mu\mathbf{I}_n & & \\ \mu\mathbf{I}_n & \ddots & \ddots & \\ & \ddots & \ddots & \mu\mathbf{I}_n \\ & & \mu\mathbf{I}_n & \mathbf{A}_t \end{pmatrix}, \tag{16}$$

while the corresponding supra-degree matrix is

$$\begin{aligned}\mathbf{D}_{\text{sup}} = \operatorname{diag}\big(&\mathbf{D}_{t-w_p+1} + \mu\mathbf{I}_n,\ \mathbf{D}_{t-w_p+2} + 2\mu\mathbf{I}_n,\ \dots,\\ &\mathbf{D}_{t-1} + 2\mu\mathbf{I}_n,\ \mathbf{D}_t + \mu\mathbf{I}_n\big).\end{aligned} \tag{17}$$

Therefore, the combinatorial supra-Laplacian $\mathbf{L}_{\text{sup}}^{\text{c}} = \mathbf{D}_{\text{sup}} - \mathbf{A}_{\text{sup}}$ has diagonal blocks $\mathbf{L}_\tau + \alpha_\tau\mu\mathbf{I}_n$ and off-diagonal blocks $-\mu\mathbf{I}_n$, where $\alpha_\tau = 1$ for the two boundary layers and $\alpha_\tau = 2$ otherwise. Explicitly,

$$\mathbf{L}_{\text{sup}}^{\text{c}} = \begin{pmatrix} \mathbf{L}_{t-w_p+1} + \mu\mathbf{I}_n & -\mu\mathbf{I}_n & & \\ -\mu\mathbf{I}_n & \ddots & \ddots & \\ & \ddots & \ddots & -\mu\mathbf{I}_n \\ & & -\mu\mathbf{I}_n & \mathbf{L}_t + \mu\mathbf{I}_n \end{pmatrix}. \tag{18}$$

Since $\mathbf{L}_{\text{sup}}$ is symmetric positive semidefinite, it is known that we can compute its k smallest eigenvectors as solutions of

$$\min_{\mathbf{U}^\top\mathbf{U}=\mathbf{I}} \operatorname{tr}\left(\mathbf{U}^\top\mathbf{L}_{\text{sup}}\mathbf{U}\right) \iff \min_{\mathbf{V}^\top\mathbf{D}_{\text{sup}}\mathbf{V}=\mathbf{I}} \operatorname{tr}\left(\mathbf{V}^\top\mathbf{L}_{\text{sup}}^{\text{c}}\mathbf{V}\right). \tag{19}$$

It remains to expand the trace according to the block structure of $\mathbf{L}_{\text{sup}}^{\text{c}}$. Since $\mathbf{L}_{\text{sup}}^{\text{c}}$ is block tridiagonal, the matrix product $\mathbf{V}^\top\mathbf{L}_{\text{sup}}^{\text{c}}\mathbf{V}$ decomposes into diagonal and adjacent off-diagonal block

multiplications. Applying the trace operator to this product yields the sum of these block-wise traces:

$$\operatorname{tr}\left(\mathbf{V}^\top \mathbf{L}_{\text{sup}}^{\text{c}} \mathbf{V}\right) = \underbrace{\sum_{\tau=t-w_p+1}^{t} \operatorname{tr}\left(\mathbf{V}_\tau^\top (\mathbf{L}_\tau + \alpha_\tau \mu \mathbf{I}_n) \mathbf{V}_\tau\right)}_{\text{diagonal blocks contribution}} - \underbrace{\mu \sum_{\tau=t-w_p+2}^{t} \left[\operatorname{tr}\left(\mathbf{V}_{\tau-1}^\top \mathbf{V}_\tau\right) + \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{V}_{\tau-1}\right)\right]}_{\text{off-diagonal blocks contribution}}. \tag{20}$$

By applying the linearity of the trace to split the first term, and using invariance of the trace under transposition $\operatorname{tr}(\mathbf{V}_\tau^\top \mathbf{V}_{\tau-1}) = \operatorname{tr}(\mathbf{V}_{\tau-1}^\top \mathbf{V}_\tau)$ to combine the cross-terms in the second, the equation simplifies to:

$$\begin{aligned}\operatorname{tr}\left(\mathbf{V}^\top \mathbf{L}_{\text{sup}}^{\text{c}} \mathbf{V}\right) = &\sum_{\tau=t-w_p+1}^{t} \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{L}_\tau \mathbf{V}_\tau\right) \\ &+ \mu \left[\sum_{\tau=t-w_p+1}^{t} \alpha_\tau \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{V}_\tau\right) - 2 \sum_{\tau=t-w_p+2}^{t} \operatorname{tr}\left(\mathbf{V}_{\tau-1}^\top \mathbf{V}_\tau\right)\right].\end{aligned} \tag{21}$$

To evaluate the term in brackets, recall that α_τ represents the temporal degree of layer τ along the path graph connecting consecutive snapshots: $\alpha_{t-w_p+1} = \alpha_t = 1$ for the boundary snapshots, and $\alpha_\tau = 2$ for every intermediate snapshot. Consequently, the diagonal sum can be partitioned over all $w_p - 1$ consecutive pairs:

$$\sum_{\tau=t-w_p+1}^{t} \alpha_\tau \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{V}_\tau\right) = \sum_{\tau=t-w_p+2}^{t} \left[\operatorname{tr}\left(\mathbf{V}_{\tau-1}^\top \mathbf{V}_{\tau-1}\right) + \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{V}_\tau\right)\right]. \tag{22}$$

Recombining this with the cross-terms and completing the square inside the trace gives

$$\begin{aligned}&\sum_{\tau=t-w_p+1}^{t} \alpha_\tau \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{V}_\tau\right) - 2 \sum_{\tau=t-w_p+2}^{t} \operatorname{tr}\left(\mathbf{V}_{\tau-1}^\top \mathbf{V}_\tau\right) \\ &\quad= \sum_{\tau=t-w_p+2}^{t} \operatorname{tr}\left(\mathbf{V}_{\tau-1}^\top \mathbf{V}_{\tau-1} + \mathbf{V}_\tau^\top \mathbf{V}_\tau - 2\mathbf{V}_{\tau-1}^\top \mathbf{V}_\tau\right) \\ &\quad= \sum_{\tau=t-w_p+2}^{t} \operatorname{tr}\left((\mathbf{V}_\tau - \mathbf{V}_{\tau-1})^\top (\mathbf{V}_\tau - \mathbf{V}_{\tau-1})\right) \\ &\quad= \sum_{\tau=t-w_p+2}^{t} \|\mathbf{V}_\tau - \mathbf{V}_{\tau-1}\|_F^2.\end{aligned} \tag{23}$$

Summing the structural and temporal contributions, we obtain

$$\operatorname{tr}\left(\mathbf{V}^\top \mathbf{L}_{\text{sup}}^{\text{c}} \mathbf{V}\right) = \sum_{\tau=t-w_p+1}^{t} \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{L}_\tau \mathbf{V}_\tau\right) \quad + \mu \sum_{\tau=t-w_p+2}^{t} \|\mathbf{V}_\tau - \mathbf{V}_{\tau-1}\|_F^2. \tag{24}$$

The first term therefore measures smoothness within each snapshot, whereas the second penalizes changes between consecutive snapshots, enforcing smooth changes on eigenvectors in consecutive layers. Substituting this decomposition into minimization objective of Equation (19) gives

$$\min_{\mathbf{V}^\top \mathbf{D}_{\text{sup}} \mathbf{V} = \mathbf{I}} \sum_{\tau=t-w_p+1}^{t} \operatorname{tr}\left(\mathbf{V}_\tau^\top \mathbf{L}_\tau \mathbf{V}_\tau\right) + \mu \sum_{\tau=t-w_p+2}^{t} \|\mathbf{V}_\tau - \mathbf{V}_{\tau-1}\|_F^2. \tag{25}$$