LEARNING RELATIONS FOR TEMPORAL GRAPH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal graph learning has traditionally relied on dataset-specific models, whereas recent advances suggest that large-scale pre-training can yield broadly transferable representations. Extending this foundation model paradigm to temporal graphs is difficult because structural and temporal information interact across heterogeneous data regimes: spatiotemporal graphs with evolving node signals and fixed topology, interaction networks with dynamic edges and static features, and fully dynamic graphs in which both change. We address this heterogeneity with a unified architecture that maps such diverse observation formats into a shared representation space. The model captures spatio-temporal patterns by integrating the observed topology with a temporal correlation-based message-passing operator. When pre-trained across multiple graphs within each domain, our model transfers zero-shot to unseen test graphs, achieving state-of-the-art performance. These results provide a concrete step toward temporal graph foundation models and highlight the benefits of treating relational structure as data-dependent.

1 INTRODUCTION

Many real-world systems evolve jointly in their relational structure and signal attributes, naturally giving rise to temporal graphs (Kazemi et al., 2020; Longa et al., 2023; Jin et al., 2024). The success of Foundation Models (FMs) in natural language processing (Brown et al., 2020; Ouyang et al., 2022) and computer vision (He et al., 2021; Kirillov et al., 2023) has sparked growing interest in Graph Foundation Models (GFMs) (Wang et al., 2025). Yet, despite rapid progress on static graphs, temporal graph learning remains dominated by a one-model-per-dataset paradigm. Current spatio-temporal neural networks and dynamic graph architectures are typically tailored to individual benchmarks (Li et al., 2018; Zhao et al., 2020; Cini et al., 2023), limiting knowledge transfer across heterogeneous temporal graph collections and hindering the development of unified, multi-domain foundation models.

A unified foundation model for temporal graphs must overcome substantial architectural and conceptual hurdles. Unlike time series modeling, where universal forecasting (Woo et al., 2024) provides a common learning objective, temporal graph datasets span three markedly different observation regimes (as shown in Figure 1): *(i)* spatio-temporal graphs with dense node time series over static topologies (*e.g.,* traffic forecasting) (Marisca et al., 2025); *(ii)* featureless interaction networks with evolving connectivity but no dynamic node attributes (Shamsi et al., 2026); and *(iii)* fully dynamic graphs in which topology and attributes evolve jointly (Na et al., 2025). Depending on the domain, key signals, such as node attributes or temporal edge trajectories, may therefore be unavailable. Input and output dimensionalities also vary drastically across datasets, complicating their processing through a unified and standardized architecture. Finally, most existing methods treat the observed topology as a complete specification of spatial dependencies, an assumption that breaks down when observed edges are noisy, incomplete, or misaligned with the underlying dynamics.

We address these challenges with **TRACE** (**T**emporal **R**elational **A**daptive **C**ontext **E**ncoder), a unified architecture designed for pre-training across heterogeneous temporal graphs. TRACE maps variable-dimensional inputs to a shared temporal representation space, accommodating graph regimes with missing dynamic attributes or structural information. Rather than relying exclusively on a fixed observed topology, TRACE composes two complementary information relational priors: the observed graph structure and a dynamic message-passing operator leveraging temporal

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.

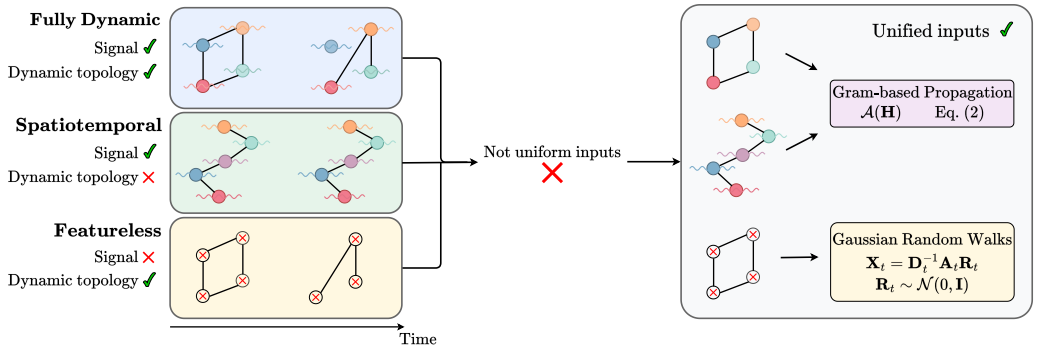


Figure 1: **Unifying heterogeneous temporal graph inputs with TRACE**. Left: possible input data scenarios when pre-training a model on snapshot-based temporal graphs, exposing heterogeneous data regimes. Right: TRACE maps these regimes into a common representation using Gaussian random-walk features for featureless graphs and Gram-based message passing.

correlations. This operator infers latent dependencies directly from temporal observations, making relational structure adaptive and data-dependent. We independently pre-train TRACE on a diverse pre-training corpus spanning three domains: traffic flow, transaction networks, and transportation networks.

Our main contributions are threefold: *(i)* we introduce TRACE, a unified architecture that processes heterogeneous temporal graph datasets across varying observation scenarios, feature dimensions, and task formulations within a shared representation space; *(ii)* we develop a novel message-passing mechanism that composes observed topology with temporal correlation-based connectivity, capturing latent interactions that emerge directly from temporal dynamics; and *(iii)* we show that TRACE generalizes zero-shot to entirely unseen test graphs across three domains without fine-tuning or retraining, while releasing our pre-trained model checkpoints for community use.

2 Related Work

Graph Foundation Models (GFMs). GFMs strive for cross-dataset generalization, yet handling heterogeneous and domain-shifting node features remains a key bottleneck. To tackle feature variation, one line of research leverages Large Language Models (LLMs) via prompt engineering, graph-to-text translation, or feature augmentation (Liu et al., 2024b; Zhao et al., 2023; Chen et al., 2024a); however, these methods can distort fine-grained topology and are largely restricted to text-attributed graphs (Wang et al., 2024; Chen et al., 2024b). Another line aligns disparate feature spaces through learned projections (Xia & Huang, 2024; Yu et al., 2024), mixture-of-experts aligners (Yu et al., 2025; Yuan et al., 2025a), or structural encodings (Frasca et al., 2025; Galkin et al., 2024). Finally, Eliasof et al. (2026) showed how to achieve transferable graph representation from input feature spaces of varying dimensionality. These methods are tailored to the static graph settings, without considering the temporal dimension. In contrast, our TRACE operates on temporal graphs, which pose different challenges than for static graphs.

Time Series Foundation Models (TSFMs). Recent work has established foundation models for zero-shot forecasting across heterogeneous time-series domains (Goswami et al., 2024; Ansari et al., 2024; Woo et al., 2024; Nie et al., 2023). These models adopt different strategies for representing temporal dynamics: patch-based approaches such as PatchTST (Nie et al., 2023) and MomentFM (Goswami et al., 2024) operate on temporally aggregated segments, while token-based models such as Chronos (Ansari et al., 2024) and TiRex (Auer et al., 2025) discretize continuous observations into tokens for sequence modeling. Recent multivariate models, including Moirai (Woo et al., 2024) and Chronos-2 (Ansari et al., 2025), further model dependencies across series, allowing interactions between entities to be learned from their observations. These approaches, however, infer such dependencies from the observed time series rather than incorporating an explicit relational structure describing how entities are connected. TRACE addresses this complementary aspect by combining

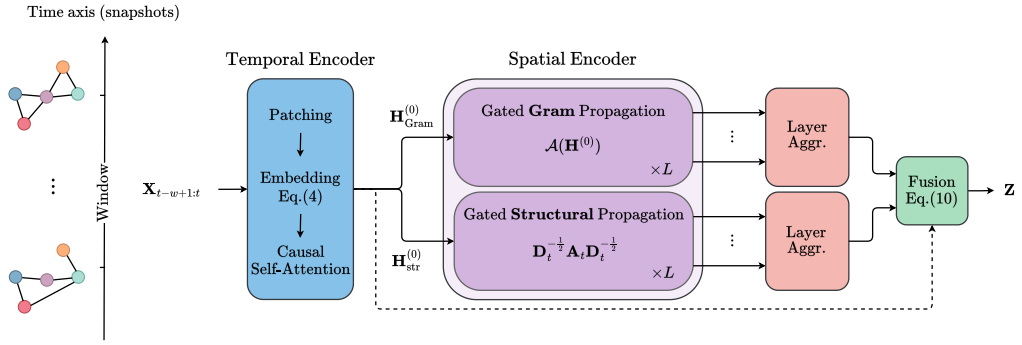


Figure 2: Overall architecture of TRACE. From left to right: after adapting heterogeneous inputs to a shared representation, a patched temporal encoder extracts node-wise dynamics independently of the graph. Two complementary message-passing branches then propagate these representations over an input-dependent Gram structure and the observed topology. Layer-wise trajectories are aggregated and fused into the final encoding.

transferable temporal representations with explicit graph structure, using a dual message-passing mechanism to model topological and dynamic dependencies across nodes.

Transferable Temporal Graph Learning. Recent efforts toward transferable temporal graph learning broadly target either spatio-temporal forecasting or dynamic interaction networks. In spatio-temporal settings, frameworks such as OpenCity (Li et al., 2025), ST-LLM (Liu et al., 2024a), and UniFlow (Yuan et al., 2025b) achieve zero-shot urban transfer via patch embeddings, LLMs, or memory banks; however, they remain tied to static topologies and specific regression tasks. Conversely, models for dynamic interaction networks address node-identity memorization (which is a key bottleneck in dynamic GNNs (Chatterjee et al., 2025)) by context-switching across datasets (Shamsi et al., 2026) or learning relative temporal embeddings (Pan et al., 2025). While these approaches mitigate identity leakage within their respective domains, they remain bounded by rigid task interfaces and still suffer from negative transfer across heterogeneous settings. In contrast, TRACE accommodates evolving topologies, varying feature dimensions, and diverse observation scenarios within a unified architecture.

3 Learning Transferable Representations

We introduce TRACE (**T**emporal **R**elational **A**daptive **C**ontext **E**ncoder), a foundational architecture for heterogeneous temporal graphs. TRACE adaptively encodes the temporal context into a unified, correlation-based representation of the underlying temporal dynamics. In Figure 2 we provide a visual representation of the overall architecture of TRACE.

3.1 Problem Formulation and Background

We jointly optimize over a set $\mathcal{D}$ of N-many Discrete-Time Dynamic Graphs (D-TDGs) (Kazemi et al., 2020; Gravina & Bacciu, 2024). Each graph is a sequence $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_T\}$ whose length T may vary across the N samples. At time index t, an observable graph snapshot $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$ comprises a non-empty node set $\mathcal{V}_t$, an edge set $\mathcal{E}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ and node features $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{V}_t| \times d_n}$. The adjacency matrix $\mathbf{A}_t \in \{0,1\}^{n \times n}$, with $n = |\mathcal{V}_t|$, encodes the edge information: $[\mathbf{A}_t]_{uv} = 1$ iff $(u, v) \in \mathcal{E}_t$, and 0 otherwise. We distinguish two common special cases: *(i)* if $\mathbf{A}_t = \mathbf{A}_1 \, \forall t$, the graph is typically termed spatio-temporal; *(ii)* if $\mathbf{X}_t = \mathbf{X}_1 \, \forall t$ it is typically termed an interaction network. When unambiguous, we omit the time index from features and adjacency matrices.

3.2 Data Heterogeneity Adaptation

Real-world D-TDGs expose markedly different information: some provide informative node features (Cini et al., 2023), topology alone (Shamsi et al., 2026), or both (Na et al., 2025). This hetero-

geneity hinders learning shared patterns across datasets. Moreover, observed topologies and node attributes may arise from imperfect data collection processes and may contain measurement noise or sampling artifacts (Rocha et al., 2017). Treating the observed graph as the sole message passing structure can therefore impose an overly restrictive inductive bias, because its edges may not capture the dependencies driving the data. This limitation is especially pronounced in spatiotemporal graphs, where a fixed topology may end up being unable to explain the evolving dependencies among node signals.

We address both forms of heterogeneity by decoupling node-representation from propagation-structure selection. When informative node features are unavailable, we derive them directly from topology. When the observed topology is fixed or insufficient to capture the relevant dependencies, we construct an input-dependent graph shift operator (GSO) (Sandryhaila & Moura, 2013) from the node representations. Together, these mechanisms enable a single backbone to process different combinations of node attributes and graph structure.

Gaussian Random Walks. When node features are absent or constant, topology is the only available signal. We therefore derive node features from the adjacency matrix, allowing the same backbone to process both topology-driven and feature-driven graphs. Specifically, we sample a Gaussian matrix $\mathbf{R}_t \in \mathbb{R}^{n\times p}$, with $[\mathbf{R}_t]_{ij} \sim \mathcal{N}(0,1)$, and define input features at time t as

$$\mathbf{X}_t = \mathbf{D}_t^{-1}\mathbf{A}_t\mathbf{R}_t, \qquad \mathbf{D}_t = \mathrm{diag}(\mathbf{A}_t\,\mathbf{1}_n). \tag{1}$$

Each node is therefore represented by the average of the Gaussian vectors assigned to its neighbors, weighted by one-step random-walk transition probabilities. The resulting p-dimensional vector compactly encodes local topology, with $p \ll n$. Because the construction depends only on $\mathbf{A}_t$ and the fixed random matrix $\mathbf{R}_t$, it introduces neither learned nor node-specific parameters and applies to arbitrary node sets.

Gram-based Propagation. The observed topology may omit dependencies present in the data. Given node features $\mathbf{H} \in \mathbb{R}^{n\times d}$, we normalize each node representation to unit ℓ_2 norm and define the input-dependent Gram operator $\mathcal{A} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times n}$ as

$$\widetilde{\mathbf{H}}_{i,:} = \frac{\mathbf{H}_{i,:}}{\|\mathbf{H}_{i,:}\|_2}, \qquad \mathcal{A}(\mathbf{H}) = \widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\top. \tag{2}$$

Each entry $\mathcal{A}_{ij}$ measures the pairwise similarity between the representations of nodes i and j, yielding a propagation structure that adapts to the observed signals rather than relying exclusively on the observed topology. Moreover, since $\mathcal{A}(\mathbf{H})$ is a Gram matrix, its rank is at most d, and its multiplication can be evaluated through the factorization $\mathcal{A}(\mathbf{H})\mathbf{X} = \widetilde{\mathbf{H}}(\widetilde{\mathbf{H}}^\top\mathbf{X})$, avoiding explicit materialization of the dense pairwise matrix.

Together, these two constructions unify the heterogeneous input regimes considered here. Informative node features are passed directly to the backbone, when available; when topology is the only signal, Equation (1) converts it into node features. In either case, the resulting representations define the same input-dependent GSO through Equation (2), providing a common propagation mechanism.

3.3 Temporal and Relational Modeling

TRACE decouples temporal representation learning from graph-dependent computation. Following the time-then-graph paradigm (Gao & Ribeiro, 2022), we first encode each node's history independently of the graph. We then propagate the resulting representation through two complementary relational structures, the observed topology and a learned Gram relation, while maintaining separate trajectories for each. Intermediate representations are aggregated into relation-specific representations which are finally fused with the original temporal representation. We detail each component below.

Patched Temporal Encoding. We use patching, which has proven effective in recent TSFMs (Liu et al., 2025; Nie et al., 2023), to capture local temporal semantics while reducing encoding cost. Given an input window $\mathbf{X} \in \mathbb{R}^{n\times w\times d_n}$, we partition the temporal axis into $P = \lceil w/q \rceil$ non-overlapping patches of q consecutive observations. Each node-channel pair is treated as an independent univariate series. Denoting the j-th patch by $\mathbf{X}_j \in \mathbb{R}^{nd_n\times q}$, we project it into two complementary d-dimensional representations and combine them through a feature-wise nonlinear interaction.

Specifically, we apply the shared linear projection

$$\left[\mathbf{V}_j \,\middle\|\, \mathbf{U}_j\right] = \mathbf{X}_j \mathbf{W}_q^\top, \tag{3}$$

where $\mathbf{W}_q \in \mathbb{R}^{2d \times q}$ is shared across patches and is independent of the number of input channels. We combine the two projections through a feature-wise nonlinear gate and add positional information:

$$\mathbf{E}_j = \mathrm{RMSNorm}\big(\mathbf{V}_j \odot \mathrm{Swish}(\mathbf{U}_j) + \mathbf{1}_{nd_n} \mathbf{p}_{P-j}^\top\big) \quad \in \mathbb{R}^{nd_n \times d}, \tag{4}$$

where $\mathbf{p}_\delta \in \mathbb{R}^d$ is the sinusoidal positional encoding (Vaswani et al., 2017) at distance δ from the most recent patch. Encoding distance from the present rather than absolute positions allows datasets to use different context window lengths while preserving a consistent representation of recency. The ordered patches are then processed by a K-layer Transformer encoder (Vaswani et al., 2017) with causal masking. Because attention acts only along the patch axis, each node-channel series is encoded independently, preserving graph-independent temporal processing, as above. We pool the encoded representations across channels to obtain one d-dimensional embedding per node and use the most recent patch representation (*i.e.,* the P-th one) as the spatial encoder's initial hidden state

$$\mathbf{H}^{(0)} = \mathrm{Pool}(\mathrm{TransformerEncoder}([\mathbf{E}_1, \ldots, \mathbf{E}_P];\, \mathbf{M})_P)\, \mathbf{W}_e \quad \in \mathbb{R}^{n \times d}, \tag{5}$$

where $\mathbf{M}$ is the causal mask over the P patch positions and $\mathbf{W}_e \in \mathbb{R}^{d \times d}$ is a learnable projection.

Multi-relation Spatial Encoding. The temporal representation from Equation (5) is graph-independent. We therefore propagate it through two complementary relations: the learned Gram relation from Equation (2), which captures similarity between node embeddings, and the observed topology, which encodes the graphs' explicit relational structure. Their corresponding GSOs are

$$\mathbf{S}_{\mathrm{gram}} = \mathcal{A}\Big(\mathbf{H}^{(0)}\Big), \qquad \mathbf{S}_{\mathrm{str}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}. \tag{6}$$

The two operators capture distinct relational information: the Gram relation connects nodes with similar learned temporal representations, whereas the topology propagates along the observed graph structure at time t. We maintain an independent representation and propagation path for each relation, initializing both from the temporal representation, $\mathbf{H}_r^{(0)} = \mathbf{H}^{(0)}$. At layer l, each relation $r \in \{\mathrm{gram}, \mathrm{str}\}$ propagates its previous representation through the corresponding GSO and a relation-specific transformation:

$$\mathbf{M}_r^{(l)} = \mathrm{ReLU}\left(\mathbf{S}_r \mathbf{H}_r^{(l-1)} \mathbf{W}_r^{(l)}\right) \quad \in \mathbb{R}^{n \times d}, \tag{7}$$

where $\mathbf{W}_r^{(l)} \in \mathbb{R}^{d \times d}$ is specific to each relation and layer. The resulting message is then incorporated into the previous representation through a relation-specific gate,

$$\mathbf{H}_r^{(l)} = \mathbf{H}_r^{(l-1)} + \sigma\left(\left[\mathbf{H}_r^{(l-1)} \,\middle\|\, \mathbf{M}_r^{(l)}\right] \mathbf{W}_{g,r}^\top\right) \odot \mathbf{M}_r^{(l)} \quad \in \mathbb{R}^{n \times d}, \tag{8}$$

where σ is the sigmoid function and $\mathbf{W}_{g,r} \in \mathbb{R}^{d \times 2d}$ is a weight matrix. By conditioning jointly on the current node embedding and incoming message, the gate modulates each relation's contribution at every node and channel. All transformation and gating parameters are shared across inputs and are independent of node and corpus identity, enabling the same propagation mechanism to transfer zero-shot.

Each message passing layer expands the receptive field: $\mathbf{H}_r^{(l)}$ incorporates information from the l-hop neighborhood while residual updates preserve shallower representations. Because useful receptive-field sizes vary with inputs, as a consequence of differences in their relational structure and scale, retaining only $\mathbf{H}_r^{(L)}$ would impose a single effective depth on every input. We instead aggregate the complete trajectory of intermediate embeddings using the layer-aggregation mechanism of Galron et al. (2026):

$$\Pi_r = \mathrm{LayerAggregation}(\mathbf{H}_r^{(0)}, \ldots, \mathbf{H}_r^{(L)}) \quad \in \mathbb{R}^{n \times d}. \tag{9}$$

We then fuse the two relation-specific representations with the original temporal representation through a shared projection

$$\mathbf{Z} = \left[\mathbf{H}^{(0)} \,\middle\|\, \Pi_{\mathrm{gram}} \,\middle\|\, \Pi_{\mathrm{str}}\right] \mathbf{W}_c^\top \quad \in \mathbb{R}^{n \times d}, \tag{10}$$

where $\mathbf{W}_c \in \mathbb{R}^{d \times 3d}$ is a weight matrix.

3.4 Computational Complexity

We now analyze the time and space complexity of TRACE.

Time Complexity. The K-layer temporal encoder processes P patches in $\mathcal{O}(nKP^2)$ operations (Vaswani et al., 2017). The L-layer structural branch has the standard MPNN complexity (Gilmer et al., 2017), which is $\mathcal{O}(L(n+|\mathcal{E}|))$. By exploiting associativity, the L-layer Gram branch requires $\mathcal{O}(Ln)$ operations. Finally, the layer aggregation of Galron et al. (2026) applies node-wise attention across layers, incurring $\mathcal{O}(n^2)$ cost. Under $K, L \ll n$, the total complexity is dominated by $\mathcal{O}(nP^2 + n^2)$.

Space Complexity. The temporal encoder stores patch representations in $\mathcal{O}(nP)$(Vaswani et al., 2017). Associative evaluation of Gram-based propagation avoids materializing the dense Gram matrix and requires only $\mathcal{O}(n)$ storage for node representations. Finally, layer aggregation stores $\mathcal{O}(nL)$ node representations (Galron et al., 2026). The overall memory complexity is therefore $\mathcal{O}(n(L+P))$.

4 Experiments

In this section, we discuss the empirical assessment of TRACE. In Section 4.1, we evaluate the in-domain generalization capabilities of our model. In Section 4.2, we show that the performance of TRACE improves in tandem with the size of the pre-training corpus. Finally, in Section 4.3, we ablate our model's architectural components to show that each one is needed to reach good transfer capabilities.

All experiments were performed on NVIDIA A100-SXM4-40GB GPUs. Models were trained using the Adam optimizer with an initial learning rate of 10^{-3} and early stopping on a macro-averaged validation loss across each validation split of the corpus. For all experiments, we adhere to the data splits and experimental protocols established in the original works. We provide details on the experimental settings in Appendix A.

4.1 In-Domain Generalization

We evaluate our method on three domains of diverse characteristics, showing that our method is effective at learning their patterns. For each domain considered, we evaluate TRACE in a zero-shot fashion without any form of fine-tuning, parameter adaptation, or linear probing. The precise construction of the zero-shot evaluation follows the experimental protocol appropriate to each domain and is described in the corresponding paragraph, together with the choice and motivation of the considered baselines. For each domain, we report the statistics of the used datasets in Appendix A.1.

Traffic Flow. We pre-train TRACE on nine traffic-flow spatio-temporal graphs at five-minute resolution: the four standard benchmarks PeMS0{3,4,7,8} (Song et al., 2020), and five additional Caltrans districts, PeMS-D{5, 6, 10, 11, 12}, obtained from the raw Performance Measurement System exports Chen et al. (2001). For all models evaluated in our pipeline, we use a context window of $w = 864$ and predict the next 12 steps, following the experimental settings of Gao et al. (2024). We evaluate TRACE in a leave-one-out (LOO) setting, training on eight graphs and evaluating zero-shot generalization on the held-out one. We compare against recent state-of-the-art TSFMs, including models with substantially larger parameter counts (see Appendix A.5) and pre-training data that include some of the target datasets.

Table 1 reports the forecasting results. TRACE achieves the best zero-shot performance across all four PeMS benchmarks, outperforming all TSFM baselines in the inductive setting. Among the TSFMs, Chronos-2-uni (Ansari et al., 2025) is the strongest baseline. Its group-attention mechanism allows the model to attend across sensor series in the input, thereby exploiting dependencies between sensors and capturing a form of relational structure without an explicit graph. This makes Chronos-2 a particularly relevant comparison for TRACE, which likewise models interactions across sensors but incorporates the observed traffic topology explicitly. Despite this, TRACE consistently achieves lower errors, suggesting that explicit relational structure provides complementary information to the dependencies inferred from the observed series. For reference, TRACE remains competitive with

Table 1: Performance Comparison with Baseline Models on traffic flow forecasting. Masked MAE (↓), RMSE (↓) and MAPE (↓, in %) on the 12-step forecasting horizon. **First**, **second**, and **third** best results are color-coded. The best supervised baseline result is reported from (Gao et al., 2024). † denotes models whose pretraining data include the target datasets.

Model	PeMS03			PeMS04			PeMS07			PeMS08		
	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓
Transductive												
Best supervised	13.80	24.43	13.96%	17.80	29.25	11.97%	18.65	31.44	7.84%	13.44	22.47	8.76%
moirai-1.1-R-base†	17.32	27.67	16.08%	23.55	36.94	15.93%	26.01	40.50	11.08%	19.02	29.57	11.75%
moirai-1.1-R-base-uni†	15.70	25.10	14.63%	21.49	33.98	14.85%	23.29	36.76	9.86%	16.82	26.48	10.38%
tirex†	14.17	22.84	13.24%	19.14	31.06	12.48%	20.04	32.95	8.32%	14.38	23.46	9.04%
flowstate†	14.30	22.96	13.57%	19.35	31.20	12.98%	20.18	32.94	8.45%	14.59	23.50	9.18%
Inductive												
chronos-2	**16.18**	**26.59**	**15.55%**	**21.92**	**35.30**	15.19%	**23.55**	**37.80**	**10.33%**	**17.43**	**28.21**	**10.92%**
chronos-2-uni	**15.95**	**26.08**	**15.10%**	**21.72**	**34.84**	**14.83%**	**23.33**	**37.25**	**10.19%**	**17.19**	**27.53**	**10.84%**
chronos-bolt-base	17.05	27.49	15.56%	22.88	36.12	**15.13%**	25.08	39.06	10.88%	18.61	29.33	11.42%
chronos-bolt-small	17.50	28.03	16.24%	23.60	36.82	16.12%	26.04	40.32	11.63%	19.12	29.69	12.01%
TRACE-zs (Ours)	**14.16**	**22.84**	**13.19%**	**19.31**	**30.92**	**12.93%**	**21.04**	**33.76**	**9.25%**	**14.89**	**23.74**	**9.68%**

the supervised STD-MAE (Gao et al., 2024) results (reported in the table as *Best supervised*), which are reported from the original study and obtained by training directly on each target dataset.

Transaction Networks. We evaluate TRACE 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. We thus exploit our Gaussian random walk features described in Section 3 as input data. We first compare the models by ranking their test AUC on each of the 20 held-out networks, 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 3, TRACE-zs achieves the best average rank (1.50), ranking first on 12 networks and second on 6. HTGN-zs follows with an average rank of 1.80, ranking first on 6 networks and second on 12, while GC-LSTM-zs has an average rank of 2.70, with its strongest relative performance limited to a small number of networks. TRACE-zs also achieves the highest average AUC, indicating that its stronger ranking is reflected in overall predictive performance rather than being driven solely by relative ordering. The results therefore show that TRACE-zs provides strong and consistent zero-shot transfer across the diverse transaction networks.

Transportation Networks. We evaluate TRACE on a subset of MOBINS (Na et al., 2025), a collection of networked time-series forecasting datasets covering human mobility across three cities in South Korea. The datasets jointly capture spatial networks, time-varying node-level features, and origin–destination (OD) movements, enabling models to account for both spatial and temporal dependencies. Following the original task, we forecast the next 7 days from a look-back window of 4 days. We employ the same TSFM baselines as for traffic flow and evaluate TRACE in a LOO setting, *i.e.,* training on two cities and evaluating zero-shot generalization on the held-out city.

Table 2 reports the results on the three cities. TRACE achieves the lowest error across Seoul, Busan, and Daegu, outperforming all TSFM baselines and the best supervised model reported by the original benchmark. Notably, most methods sharing an inductive setting also outperform the best supervised baseline (which is reported from the original work) on all three cities, despite the zero-shot evaluation. TRACE further improves over the strongest TSFM on each city, with the largest margin on Busan.

4.2 Effect of Pre-training Corpus Size

To complement our main experiments, we examine how the number of training networks affects transfer performance on traffic forecasting. We progressively increase the number of

Table 2: Zero-shot performance (MAE ↓) on transportation forecasting benchmarks. **Best**, **second** and **third** best performances are color-coded.

Model	Seoul	Busan	Daegu
Transductive			
Best supervised	0.3822	0.5743	0.4677
Inductive			
chronos-2	0.3425	0.4965	0.4308
chronos-2-uni	**0.3321**	**0.4911**	**0.4295**
chronos-bolt-base	0.3422	0.4979	0.4391
chronos-bolt-small	0.3397	0.4947	0.4353
moirai-1.1-R-base	0.3971	0.5592	0.4755
moirai-1.1-R-base-uni	0.3925	0.5559	0.4772
tirex	**0.3329**	**0.4863**	**0.4246**
flowstate	0.3487	0.4992	0.4448
TRACE-zs (Ours)	**0.3311**	**0.4693**	**0.4119**

Table 3: Zero-shot performance (AUC ↑) on 20 MiNT Test Networks, averaged over 3 seeds. **Best** and **second** best performances are color-coded.

Network	GC-LSTM-zs	HTGN-zs	TRACE-zs (Ours)
MIR	0.698 ± 0.021	**0.720 ± 0.015**	**0.730 ± 0.015**
DOGE2.0	**0.551 ± 0.022**	0.538 ± 0.038	**0.718 ± 0.080**
MUTE	0.610 ± 0.020	**0.673 ± 0.013**	**0.685 ± 0.013**
EVERMOON	0.502 ± 0.011	**0.517 ± 0.039**	**0.832 ± 0.028**
DERC	0.642 ± 0.018	**0.798 ± 0.027**	**0.770 ± 0.006**
ADX	0.677 ± 0.018	**0.679 ± 0.024**	**0.694 ± 0.008**
HOICHI	**0.847 ± 0.005**	**0.765 ± 0.018**	0.640 ± 0.015
SDEX	**0.762 ± 0.034**	0.614 ± 0.020	**0.730 ± 0.169**
BAG	0.610 ± 0.030	**0.931 ± 0.028**	**0.906 ± 0.038**
XCN	0.605 ± 0.022	**0.851 ± 0.043**	**0.707 ± 0.006**
ETH2x-FLI	0.610 ± 0.059	**0.729 ± 0.015**	**0.774 ± 0.006**
stkAAVE	0.640 ± 0.018	**0.709 ± 0.022**	**0.753 ± 0.017**
GLM	0.780 ± 0.012	**0.831 ± 0.024**	**0.895 ± 0.014**
QOM	0.595 ± 0.015	**0.647 ± 0.019**	**0.692 ± 0.029**
WOJAK	0.450 ± 0.008	**0.524 ± 0.027**	**0.700 ± 0.020**
DINO	0.403 ± 0.021	**0.779 ± 0.113**	**0.860 ± 0.008**
Metis	0.680 ± 0.019	**0.760 ± 0.025**	**0.712 ± 0.020**
REPv2	0.700 ± 0.011	**0.789 ± 0.020**	**0.804 ± 0.008**
TRAC	**0.670 ± 0.020**	**0.785 ± 0.008**	0.665 ± 0.024
BEPRO	0.620 ± 0.015	**0.782 ± 0.003**	**0.653 ± 0.026**
Avg. AUC ↑	0.633	0.721	**0.746**
Avg. Rank ↓	2.70	1.80	**1.50**

PeMS networks used for training and evaluate the resulting model on each held-out network. Specifically, $\mathcal{D}_1 = \{\text{PeMS-D5}\}$, $\mathcal{D}_2$ additionally includes PeMS-D6, $\mathcal{D}_4$ further adds PeMS-D10 and PeMS-D11, and $\mathcal{D}_8$ includes all networks of Table 1 while leaving the target network out for zero-shot evaluation. Finally, $\mathcal{D}_9$ uses all nine networks, including the target network, and therefore corresponds to transductive rather than zero-shot evaluation. The resulting performance across these training configurations is shown in Figure 3. As the number of training networks increases, forecasting error consistently decreases across the traffic flow networks overall, with the benefits becoming increasingly pronounced as more diverse networks are included. This trend shows that exposing TRACE to a broader range of network structures and temporal patterns improves its transfer to unseen networks within the same domain. The final zero-shot setting, where the target network is excluded from training, provides strong improvements across all target networks, demonstrating that the scaling benefits extend beyond the networks observed during training. Including the target network during training provides a further improvement, as expected, since the model can directly exploit its specific patterns. This transductive setting also brings the shared model close to the performance of specialist models trained separately for each network.

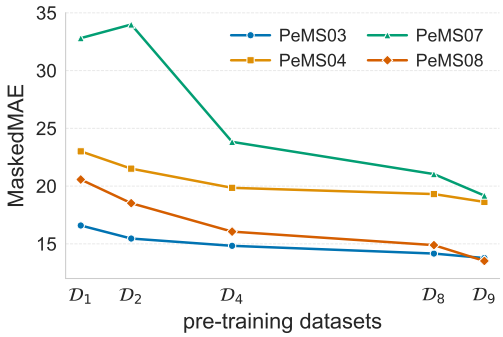


Figure 3: TRACE performance generally improves with a larger pre-training corpus.

4.3 Ablation Study

We isolate the contribution of the main components of TRACE by progressively removing or modifying one component at a time, while keeping all remaining components and hyperparameters unchanged. The ablations aim at evaluating three aspects of the model: the role of relational propagation, the complementarity of the learned and observed relations, and the design of the learned representations and their aggregation across propagation depths. We perform all ablations using the pre-trained weights of the full model, which are kept frozen during evaluation without any target-corpus adaptation. This evaluation protocol reflects the purpose of this analysis, which is showing that every component of our architecture is needed to have good zero-shot transfer.

Table 4: **Ablation.** Masked MAE (↓) on the test split of each held-out flow corpus, averaged over the 12 prediction steps. **First**, **second** and **third** best are color-coded.

Variant	Relaxes	PeMS03	PeMS04	PeMS07	PeMS08	Mean
TRACE (full)	—	**14.16**	**19.31**	**21.04**	**14.89**	**17.35**
no propagation	Equation (7)	18.91	22.68	26.70	19.70	22.00
Gram relation only	Equation (6)	15.77	19.91	22.50	15.34	**18.38**
topology only	Equation (6)	16.15	21.12	24.39	18.00	**19.91**
Gram from $\mathbf{X}$	Equation (2)	21.16	26.87	34.48	24.98	26.87
no layer aggregation	Equation (9)	19.94	25.50	29.22	20.64	23.82

Relational propagation. We first assess the role and complementarity of the two relational pathways. **No propagation** removes the message in Equation (7), so that each node is encoded independently and the model reduces to a per-series forecaster. **Gram relation only** retains the learned relation in Equation (6) while removing the observed topology, whereas **topology only** retains the observed topology and removes the Gram relation, corresponding to the relational structure used by conventional spatiotemporal models.

Relation construction and aggregation. We then assess how the learned relation is constructed and how information across propagation depths is retained. **Gram from $\mathbf{X}$** constructs the affinity in Equation (2) from the raw input window rather than the learned state $\mathbf{H}^{(0)}$. Finally, **no layer aggregation** uses only the deepest representation $\mathbf{H}_r^{(L)}$ instead of the trajectory collected in Equation (9), thereby fixing the effective propagation depth across different datasets.

Table 4 reports the results of the ablations. The full TRACE configuration achieves the lowest mean error. Removing propagation leads to a substantial degradation, showing that propagating information through the learned relations and observed topology contributes to performance. Separating the two relation components also degrades performance, with using only the Gram relation performing better than using only the observed topology. Constructing the Gram relation directly from the raw input $\mathbf{X}$ produces a further substantial degradation, highlighting the importance of the temporal representations for estimating the learned relations. Finally, removing layer aggregation increases the error, indicating that retaining representations from multiple propagation depths is beneficial.

5 Conclusions

We introduced TRACE, a unified model for heterogeneous temporal graphs. TRACE maps temporal graph datasets with different feature dimensions and observation regimes into a shared representation space, enabling pre-training across spatiotemporal, interaction, and fully dynamic graphs. Its relational message-passing mechanism combines observed topology with temporal correlation-based connectivity, allowing the model to capture dependencies that are not fully specified by the observed graph structure. Across traffic flow, transaction, and transportation networks, TRACE transfers zero-shot to unseen graphs and achieves state-of-the-art performance across the evaluated settings. Ablation and scaling analyses further support the contribution of the learned relational structure and the value of diverse pre-training data.

These results demonstrate the feasibility of pre-training a single model across heterogeneous temporal graph domains and transferring the resulting representations to previously unseen graphs without dataset-specific adaptation. Future work includes scaling joint pre-training to larger and more diverse temporal graph corpora and extending the framework to continuously sampled dynamic graphs.

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 TRACE 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.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=v7UqniC9pF.

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.

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.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (llms)in learning on graphs. *SIGKDD Explor. Newsl.*, 25(2):42–61, March 2024a. ISSN 1931-0145. doi: 10.1145/3655103.3655110. URL https://doi.org/10.1145/3655103.3655110.

Zhikai Chen, Haitao Mao, Jingzhe Liu, Yu Song, Bingheng Li, Wei Jin, Bahare Fatemi, Anton Tsitsulin, Bryan Perozzi, Hui Liu, and Jiliang Tang. Text-space graph foundation models: Comprehensive benchmarks and new insights. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=Q2sDuwtutB.

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.

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.

Moshe Eliasof, Krishna Sri Ipsit Mantri, Beatrice Bevilacqua, Bruno Ribeiro, and Carola-Bibiane Schönlieb. Bridging input feature spaces towards graph foundation models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Dt4XAIKYbf.

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.

Fabrizio Frasca, Fabian Jogl, Moshe Eliasof, Matan Ostrovsky, Carola-Bibiane Schönlieb, Thomas Gärtner, and Haggai Maron. Towards foundation models on graphs: An analysis on cross-dataset transfer of pretrained GNNs. In *NeurIPS 2024 Workshop on Symmetry and Geometry in Neural Representations*, 2025. URL https://openreview.net/forum?id=URMPK7Yp9Y.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 31598–31619, 2024.

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*, 2026. URL https://openreview.net/forum?id=8SnAGYf2wM.

Haotian Gao, Renhe Jiang, Zheng Dong, Jinliang Deng, Yuxin Ma, and Xuan Song. Spatial-temporal-decoupled masked pre-training for spatiotemporal forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/442.

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.

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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

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.

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.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

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.

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.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting, 2023.

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*, 2024a.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=4IT2pgc9v6.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=SrEOUSyJcR.

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.

Ivan Marisca, Jacob Bamberger, Cesare Alippi, and Michael M. Bronstein. Over-squashing in spatiotemporal graph neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=CVp0WCw4a1.

Jihye Na, Youngeun Nam, Susik Yoon, Hwanjun Song, Byung Suk Lee, and Jae-Gil Lee. Mobility networked time-series forecasting benchmark datasets. In *ICWSM*, pp. 2539–2549, 2025. URL https://doi.org/10.1609/icwsm.v19i1.35955.

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.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

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.

Luis E. C. Rocha, Naoki Masuda, and Petter Holme. Sampling of temporal networks: Methods and biases. *Phys. Rev. E*, 96:052302, Nov 2017. doi: 10.1103/PhysRevE.96.052302. URL https://link.aps.org/doi/10.1103/PhysRevE.96.052302.

Aliaksei Sandryhaila and José M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013. doi: 10.1109/TSP.2013.2238935.

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.

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.

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.

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.

Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. GFT: Graph foundation model with transferable tree vocabulary. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=0MXzbAv8xy.

Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, Qingkai Zeng, Xiusi Chen, Jianan Zhao, Jundong Li, Meng Jiang, Pietro Lio, Nitesh Chawla, Chuxu Zhang, and Yanfang Ye. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025.

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.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2023.

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.

Lianghao Xia and Chao Huang. Anygraph: Graph foundation model in the wild. *arXiv preprint arXiv:2408.10700*, 2024.

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.

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.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*, 2024.

Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. Samgpt: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 1142–1153, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746. doi: 10.1145/3696410.3714828. URL https://doi.org/10.1145/3696410.3714828.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S. Yu. How much can transfer? BRIDGE: Bounded multi-domain graph foundation model with generalization guarantees. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73604–73644. PMLR, 13–19 Jul 2025a.

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*, 2025b.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504*, 2022.

Haiteng Zhao, Shengchao Liu, Chang Ma, Hannan Xu, Jie Fu, Zhi-Hong Deng, Lingpeng Kong, and Qi Liu. Gimlet: a unified graph-text model for instruction-based molecule zero-shot learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

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.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *arXiv preprint arXiv:2012.07436*, 2021.

A Experimental Details

A.1 Employed Datasets

We report the statistics of every corpus employed in this work, grouped by domain it belongs to. For every corpus we adhere to the split of the original work. The corpora span five domains: traffic flow, social interactions, transaction, transportation and epidemic spreading.

A.1.1 Traffic Flow

We consider nine traffic-flow forecasting corpora derived from the California Performance Measurement System (PeMS) (Chen et al., 2001). PeMS records loop-detector measurements of the number of vehicles observed at five-minute intervals, yielding 12 observations per hour. We use the four standard benchmarks PeMS03,4,7,8 and five additional districts, PeMS-D5,6,10,11,12, obtained from the raw PeMS Data Clearinghouse. Each corpus is represented as a graph whose nodes correspond to detectors and whose edges encode their spatial connectivity. Table 5 reports the statistics and sources of all nine corpora.

For PeMS03,4,7,8, we use the datasets and road adjacencies released with the original benchmarks. For the five additional districts, we construct the datasets from the PeMS Data Clearinghouse (Chen et al., 2001). We use the total-flow measurement of each detector and discard detectors observed in fewer than 90% of snapshots. Since the Clearinghouse provides detector coordinates but no road-network distances, we construct the adjacency using the Gaussian kernel adopted by Li et al. (2018); Yu et al. (2018), with edge weights $w_{ij} = \exp(-d_{ij}^2/\sigma^2)$, with d_{ij} being the distance between detectors i and j and σ is the kernel bandwidth.

The task is traffic-flow forecasting: given a historical window of observations from all detectors, models predict the next $H = 12$ steps, corresponding to one hour. We follow the chronological $6 : 2 : 2$ train–validation–test split of the original benchmark (Guo et al., 2019) and use a single input channel for all datasets and methods (*i.e.,* we do not include any other additional inputs such as covariates). For the zero-shot experiments, we adopt a leave-one-out protocol, pre-training on eight datasets, with context window corresponding to three days of observations (matching the setting of Gao et al. (2024)) and evaluating on the held-out corpus.

Table 5: **Traffic flow datasets.** $|\mathcal{V}|$ is the number of loop detectors, $|\mathcal{E}|$ the number of edges in the adjacency each corpus supplies, and T the number of five-minute snapshots. Every corpus carries a single input channel, *i.e.,* the vehicle count at a detector, and every model predicts $H = 12$ steps, *i.e.,* one hour ahead.

dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	T	split	source
PeMS03	358	1 093	26 208	6:2:2	(Guo et al., 2019)
PeMS04	307	209	16 992	6:2:2	(Guo et al., 2019)
PeMS07	883	790	28 224	6:2:2	(Guo et al., 2019)
PeMS08	170	137	17 856	6:2:2	(Guo et al., 2019)
PeMS-D5	150	1 200	25 045	6:2:2	Clearinghouse D5
PeMS-D6	471	3 768	25 045	6:2:2	Clearinghouse D6
PeMS-D10	490	3 920	25 045	6:2:2	Clearinghouse D10
PeMS-D11	702	5 616	25 045	6:2:2	Clearinghouse D11
PeMS-D12	820	6 560	25 045	6:2:2	Clearinghouse D12

A.1.2 Transaction Networks

We utilize the benchmark collection of 84 temporal transaction networks introduced by the Temporal Multi-network Transfer (MiNT) framework (Shamsi et al., 2026). These datasets represent Ethereum-based cryptocurrency token transactions, collectively comprising approximately 3 million nodes and 19 million temporal edges. The primary task on these datasets is temporal graph property prediction, which requires models to forecast structural characteristics or aggregated activity levels of the network over discrete temporal snapshots.

Following the original MiNT evaluation protocol, we adopt a multi-network transfer learning setting to test zero-shot generalization. Models are pre-trained on a diverse subset of up to 64 token networks. Subsequently, their transferability is evaluated on the remaining 20 held-out, strictly unseen token networks. We follow the original work settings and perform a temporal training–validation–test split of $70:15:15$, for each considered network.

The statistics of the 84 employed token networks are shown in Table 6.

Table 6: 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.1.3 TRANSPORTATION

We utilize three transportation datasets from the MOBINS benchmark (Na et al., 2025), which record public-transport movements across three metropolitan areas in South Korea: Seoul, Busan, and Daegu. The data is sampled at an hourly rate, and each corpus is represented as a dynamic graph where nodes correspond to administrative areas. For the input features, each node supplies two time-series signals ($d_n = 2$) representing the hourly subway inflow and outflow. Crucially, these datasets exhibit a time-varying origin-destination (OD) adjacency matrix, where each entry encodes the number of smart-card users traveling from an origin to a destination area. The primary task on these datasets is spatio-temporal forecasting: given a historical context window of 4 days of hourly observations (comprising both the node-level features and the dynamic $n \times n$ OD topology), models must forecast these exact same targets (*i.e.,* inflow, outflow, and the OD matrix) for the future 7 days. The number of targets is therefore $d_n|\mathcal{V}| + |\mathcal{V}|^2$. We strictly adhere to the chronological temporal data splits established in the original benchmark, which is $60:15:25$ for train, validation, and test sets.

Table 7: **MOBINS transportation benchmark datasets.** $|\mathcal{V}|$ is the number of regions, $|\mathcal{E}|$ the number of directed edges in the static spatial network the benchmark supplies, and T the number of snapshots, hourly.

corpus	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	T	targets	split
Seoul	128	324	17 520	16 640	60 : 15 : 25
Busan	60	122	26 280	3 720	60 : 15 : 25
Daegu	61	124	26 280	3 843	60 : 15 : 25

A.2 Employed Baselines

To evaluate the effectiveness of TRACE, we compare it against a comprehensive set of baselines. We divide them by supervised models and TSFMs below:

Supervised Models. **AGCRN** (Bai et al., 2020), **ASTGCN** (Guo et al., 2019), **ASTGNN** (Guo et al., 2022), **DCRNN** (Li et al., 2018), **DSTAGNN** (Lan et al., 2022), **EnhanceNet** (Cirstea et al., 2021), **GC-LSTM** (Chen et al., 2022), Graph WaveNet (**GWNet**) (Wu et al., 2019), **HTGN** (Yang et al., 2021), **MPNN-LSTM** (Gilmer et al., 2017), **PDFormer** (Jiang et al., 2023), **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** (Song et al., 2020), **ST-WA** (Cirstea et al., 2022), **Transformer** (Vaswani et al., 2017), **Z-GCNETs** (Chen et al., 2021). **DLinear** (Zeng et al., 2022), **NLinear** (Zeng et al., 2022), **SegRNN** (Lin et al., 2023), **Informer** (Zhou et al., 2021), **Reformer** (Kitaev et al., 2020), **PatchTST** (Nie et al., 2023), **TimesNet** (Wu et al., 2023),

Time Series Foundation Models. `moirai-1.1-R-base` (Woo et al., 2024), `moirai-1.1-R-base-uni` (Woo et al., 2024), `tirex` (Auer et al., 2025), `flowstate` (Graf et al., 2026), `chronos-2` (Ansari et al., 2025), `chronos-2-uni` (Ansari et al., 2025), `chronos-bolt-base` (Ansari et al., 2024), `chronos-bolt-small` (Ansari et al., 2024).

A.3 Implementation Details

Joint Pre-training. Our pre-training sampling mechanism accounts for the heterogeneity of the graphs, which may differ in number of nodes, edges, sequence length and number of input features. At each step, we first sample a dataset from the pool of training datasets and then sample a batch of windows from it. To prevent larger datasets from dominating training, we allocate a fixed capacity of batches, Cap, to each dataset within an epoch. Batches are sampled without replacement while sufficient data remain and the sampling is cycled when the available data are exhausted.

We optimize a common loss for each dataset, but its magnitude is not directly comparable across datasets with different input scales and task objectives. We therefore normalize the loss for each dataset by the loss of a trivial baseline predictor. For forecasting, the persistence baseline simply predicts that each future value will remain equal to the most recently observed value. For classification, the baseline predicts each class according to its empirical frequency in the training set, *i.e.,* it uses the class distribution observed during training as its prediction. The baseline loss is computed using the same loss function as the model. We then divide each dataset's model loss by the corresponding baseline loss and select the best checkpoint by early stopping on the macro-averaged normalized loss across all validation datasets.

Output Heads. We tailor the readout to the specific prediction task. For forecasting, we use a node-wise readout that maps each node representation to its H future steps. For transportation datasets, we additionally use a low-rank bilinear decoder to model origin-destination interactions, in parallel with the node-wise readout. Both readouts receive a skip connection from the initial input window. For graph-level classification (*i.e.,* on the MiNT zero-shot benchmark), we average the learned representations across all node slots to obtain a graph-level representation, which is then passed to a linear classification head.

Normalization. We employ instance normalization to remove differences in signal magnitude across nodes, channels, and windows, allowing the shared model to focus on temporal patterns rather than absolute scale. Each input window is therefore z-score normalized independently per node and channel before patching. At the dataset level, we apply domain-specific normalization: for traffic forecasting, we z-score normalize using the mean and standard deviation computed on the training split, and transform predictions back to the original scale before evaluation (Gao et al., 2024); for transportation, we apply per-feature standardization fitted on the pre-test period and evaluate in the resulting scaled units following the MOBINS protocol Na et al. (2025); transaction data are not additionally normalized, following Shamsi et al. (2026).

Gram. The Gram graph is constructed only over nodes active in each window rather than over the full set of nodes. During training, we further limit the number of nodes used to construct the Gram graph to at most 4096. We retain the top-k active nodes for building the Gram matrix, while inactive

Table 8: Hyperparameter used in the experiments.

Hyperparameter	Traffic	Transportation	Transactions
Gaussian RW feature dimension p	✗	✗	16
Hidden channels d	128	128	64
Spatial Encoder layers L	4	4	4
Temporal Encoder layers K	2	2	2
Patch length q	12	24	4
Context window w	864	96	48
Weight decay	0	0	10^{-4}
Epochs	400	300	200
Learning-rate schedule	✗	✗	cosine annealing

Table 9: **Model size.** Number of parameters of TRACE and of every time-series foundation model we compare against. For TRACE we report one model per domain.

Model	Domain / release	Parameters
TRACE	Transactions	0.25 M
TRACE	Traffic flow	1.00 M
TRACE	Transportation	1.46 M
TSFMs		
flowstate	`ibm-research/flowstate`	18.5 M
tirex	`NX-AI/TiRex`	35.3 M
chronos-bolt-small	`amazon/chronos-bolt-small`	47.7 M
moirai-1.1-R-base	`Salesforce/moirai-1.1-R-base`	91.4 M
chronos-2	`amazon/chronos-2`	119.5 M
toto-open-base-1.0	`Datadog/Toto-Open-Base-1.0`	151.3 M
chronos-bolt-base	`amazon/chronos-bolt-base`	205.3 M

nodes bypass this message-passing step. To avoid division by zero, we set to zero the representation of nodes whose ℓ_2-norm over channels is close to zero.

A.4 Hyperparameter

The hyperparameter space employed by TRACE across all tasks is reported in table 8.

A.5 Model and Baseline Sizes

We report in Table 9 the number of parameters of each TSFMs we compared to across experiments, showing that the biggest version of TRACE has $\approx 12\times$ less parameters than the smallest competitor.