FROM TEMPORAL PRIORS TO SPATIAL REASONING: GRAPH ADAPTATION OF FROZEN TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent spatio-temporal foundation models (STFMs) have demonstrated strong transferability and forecasting capabilities in zero-shot and few-shot settings, but they usually require large-scale spatio-temporal pre-training and substantial computational resources. Moreover, spatial dependencies are often highly domain-specific, making it difficult for a pre-trained spatio-temporal backbone to generalize across heterogeneous spatial structures. To address these limitations, we propose Graft, a framework that integrates lightweight graph adaptation into frozen time-series foundation models (TSFMs), extending them to downstream spatio-temporal forecasting without modifying the foundation backbone. Specifically, Graft extracts latent forecast tokens from the frozen TSFM while constructing temporally aligned relational graph tokens from historical observations, node attributes, and target-domain graph structures. The graph tokens are projected into the forecast-token space and adaptively integrated with the forecast tokens through gated temporal-graph fusion. We further develop a causal-mediated graph adapter (CMGA) that models hierarchical graph dependencies and dynamically injects graph-structured corrections into the temporal representations. Extensive experiments on seven spatio-temporal forecasting benchmarks demonstrate that the proposed method achieves strong few-shot performance and favorable efficiency compared with full TSFM fine-tuning. Further experiments on multivariate time-series anomaly detection validate its generalization across different graph semantics.

1 INTRODUCTION

Spatio-temporal forecasting plays a pivotal role in a broad range of real-world applications, including traffic management, urban computing, environmental monitoring, and energy systems Li et al. (2018); Zhang et al. (2017). In recent years, the field of spatio-temporal forecasting has pursued not only higher predictive accuracy but also stronger cross-scenario generalization across diverse cities, sensing networks, and application settings Liang et al. (2025; 2024). This trend has motivated a new line of research toward pre-trained time-series or spatio-temporal foundation models, which aim to move beyond dataset-specific architectures and provide more transferable priors for downstream prediction tasks. Recent representative efforts, such as UniST, OpenCity, and FactoST, have shown that pre-training at scale can substantially improve the universality and zero-shot and few-shot capabilities of spatio-temporal prediction models Yuan et al. (2024)–Zhong et al. (2025).

Despite this promise, a natural solution, pre-training a dedicated spatio-temporal foundation model from scratch, is still far from ideal. As shown in Figure 1, joint spatio-temporal pre-training usually requires large-scale multi-domain data collection, complex modeling of both temporal dynamics and spatial structures, and substantial computational resources for optimization and deployment. More importantly, spatio-temporal data are intrinsically heterogeneous across domains: while temporal patterns often exhibit recurring dynamics that can transfer across datasets, spatial dependencies are heavily shaped by domain-specific factors such as road topology, sensor layout, functional regions, and local interaction mechanisms. This makes end-to-end spatio-temporal foundation modeling expensive not only in computation, but also in representation design, because a single jointly pre-trained model must simultaneously absorb universal temporal regularities and highly diverse spatial semantics. Moreover, it also risks entangling transferable temporal priors with domain-specific

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.

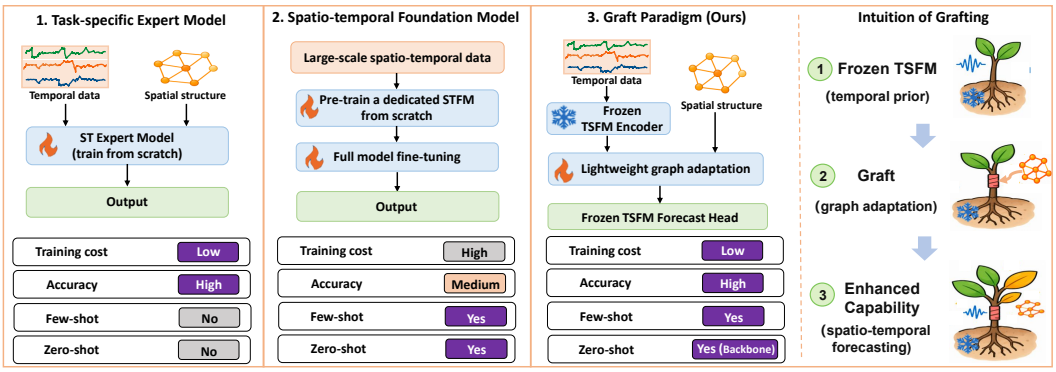


Figure 1: Comparison of existing spatio-temporal forecasting paradigms and the intuition of Graft.

spatial semantics. Such pre-trained spatial knowledge, once baked into the model backbone, may introduce biases when transferred to unseen target domains, thereby interfering with the learning of target spatial structures and potentially limiting downstream spatial adaptation. As a result, the zero-shot forecasting capability of a dedicated spatio-temporal foundation model may not always surpass that of a strong time-series foundation model (TSFM), especially when the latter has learned more generalizable temporal priors from large-scale heterogeneous time-series corpora Zhong et al. (2025). Recent studies on universal spatio-temporal foundation modeling have also highlighted the difficulty of jointly accommodating heterogeneous spatial structures within a single pre-trained model Zhong et al. (2026).

These observations suggest a more practical and principled alternative. Instead of learning a monolithic spatio-temporal foundation model from scratch, one may start from a strong TSFM and adapt it to downstream spatio-temporal forecasting through lightweight graph adaptation. This perspective is based on a natural decomposition of the problem: temporal dynamics often exhibit broadly transferable patterns that can be learned from large-scale heterogeneous time-series corpora, whereas graph-structured dependencies in spatio-temporal systems are strongly shaped by the target domain. From this viewpoint, the frozen TSFM can serve as a reusable temporal backbone, while target-specific spatial relations are introduced through lightweight downstream graph modules. Recent advances in TSFMs such as Chronos, TimesFM, Lag-Llama, and Moirai further support this idea by demonstrating that strong temporal priors can be learned once and transferred across diverse forecasting scenarios Ansari et al. (2024)–Jin et al. (2024). Meanwhile, recent studies on factorized spatio-temporal foundation modeling have shown that separating temporal representation learning from spatial structure adaptation can provide a more scalable way to handle heterogeneous spatial domains Zhong et al. (2025), Zhong et al. (2026). Taken together, these developments suggest that competitive spatio-temporal forecasting does not necessarily require jointly pre-training temporal and spatial knowledge in a dedicated foundation model. Instead, a frozen temporal foundation backbone can be coupled with lightweight target-domain graph adaptation, thereby preserving transferable temporal priors while flexibly incorporating domain-specific spatial dependencies with substantially reduced downstream data and memory requirements.

In this paper, we propose **Graft**, **a framework that integrates lightweight graph adaptation into off-the-shelf pre-trained TSFMs, extending them to downstream spatio-temporal forecasting.** Instead of pre-training a dedicated spatio-temporal foundation model or fine-tuning the time-series foundation backbone, Graft keeps the TSFM completely frozen and trains only the lightweight graph adaptation modules Houlsby et al. (2019). Specifically, we extract latent forecast tokens from the frozen TSFM while jointly modeling historical observations, node attributes, and target-domain graph structures, so that the representations of graph-structured dependencies are modulated according to the observed temporal context, yielding temporally aligned relational graph tokens. The graph tokens are then projected into the forecast-token space and adaptively integrated with the forecast tokens through gated temporal-graph fusion. To further refine structured dependencies, we develop a causal-mediated graph adapter (CMGA) that decomposes graph interactions into three complementary pathways: direct local effects, mediated regional effects, and global contextual effects. A dynamic routing mechanism adaptively balances these pathways, while a structural-strength modulation gate controls the intensity of graph-structured corrections injected into the fused temporal

representations. The adapted representations are finally decoded by the original frozen forecasting head of the TSFM, thereby providing a plug-and-play and efficient solution for downstream spatio-temporal forecasting. Although Graft is primarily developed for spatio-temporal forecasting, its graph adaptation mechanism is not restricted to physical spatial topology. We further evaluate it on multivariate time-series anomaly detection, where spatial topology is replaced by inter-variable dependency graphs, and forecasting is replaced by reconstruction-based detection. This provides a cross-task generalization test across different graph semantics and downstream objectives.

- We propose Graft, a framework that integrates lightweight target-domain graph adaptation into off-the-shelf pre-trained TSFMs, extending them to downstream spatio-temporal forecasting. Unlike approaches that rely on large-scale spatio-temporal pre-training to build dedicated STFMs or fine-tune the full foundation backbone, Graft keeps the TSFM completely frozen and trains only the lightweight graph adaptation modules.
- We develop a Causal-Mediated Graph Adapter (CMGA), which models graph interactions through direct local, mediated regional, and global contextual effects. Combined with temporally aligned relational graph tokens and gated temporal-graph fusion, CMGA enables adaptive injection of graph-structured information into frozen TSFM representations.
- Experiments demonstrate strong few-shot performance and efficient adaptation for spatio-temporal forecasting, while additional anomaly detection experiments validate generalization across different graph semantics and downstream objectives.

2 Related Work

Spatio-temporal forecasting. Task-specific spatio-temporal models combine graph neural networks with recurrent, convolutional, or attention-based temporal modules. Representative methods include DCRNN Li et al. (2018), STGCN Yu et al. (2018), Graph WaveNet Wu et al. (2019), AGCRN Bai et al. (2020), MTGNN Wu et al. (2020), PDFormer Jiang et al. (2023), and STAEformer Liu et al. (2023). These models achieve strong in-domain accuracy but are typically trained from scratch for each target dataset, require sufficient labeled data for effective training, and remain closely tied to dataset-specific graph structures.

Time-series foundation models. TSFMs learn transferable temporal priors from large heterogeneous corpora. Chronos Ansari et al. (2024), TimesFM Das et al. (2024), Lag-Llama Rasul et al. (2023), Moirai Woo et al. (2024), MOMENT Goswami et al. (2024), Timer Liu et al. (2024), and Time-LLM Jin et al. (2024) demonstrate broad transfer across forecasting and time-series analysis tasks. However, most TSFMs do not natively incorporate explicit target-domain graph structures and, when directly applied to spatio-temporal forecasting, often underperform task-specific expert models that explicitly exploit spatial dependencies.

Spatio-temporal foundation models. STEP Shao et al. (2022a), UniST Yuan et al. (2024), and OpenCity Li et al. (2025) extend large-scale pre-training to spatio-temporal prediction. FactoST Zhong et al. (2025) and FactoST-v2 Zhong et al. (2026) further emphasize temporal–spatial factorization to improve transfer across heterogeneous spatial domains. Although existing STFMs have demonstrated promising transferability across spatio-temporal domains, they typically require large-scale spatio-temporal pre-training over substantial amounts of data, leading to considerable training and deployment costs. Moreover, because spatial dependencies are highly domain-specific, the transfer of pre-trained spatial priors to unseen target domains can be limited.

3 Method

3.1 Overview

Let $\mathcal{F}_\Theta = \mathcal{D}_\Theta \circ \mathcal{E}_\Theta$ denote a pre-trained TSFM with frozen parameters Θ, where $\mathcal{E}_\Theta$ and $\mathcal{D}_\Theta$ denote its encoder and native forecasting head, respectively. For the considered tasks, each graph node corresponds to a univariate temporal sequence, so we omit the feature dimension and denote an input batch by $\mathbf{X} \in \mathbb{R}^{B\times N\times L}$. Given prediction horizon H, the frozen TSFM encoder produces

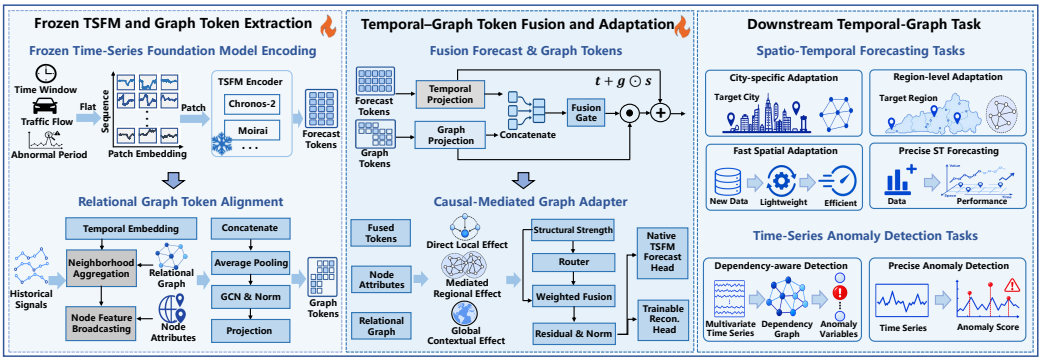


Figure 2: Overview of Graft. Graft augments frozen TSFM forecast tokens with target-domain relational graph tokens through gated temporal-graph fusion and CMGA. The adapted representations are decoded by the original frozen forecasting head for spatio-temporal forecasting, while a lightweight reconstruction head is used for anomaly detection to evaluate cross-task generalization.

future-oriented forecast tokens

$$\mathbf{Z}^T = \mathcal{E}_\Theta(\mathbf{X}) \in \mathbb{R}^{B\times N\times P\times D}, \tag{1}$$

where $P = \lceil H/S \rceil$ is the number of output patches, S is the output patch size, and D is the hidden dimension. Graft learns a lightweight graph adaptation function

$$\widetilde{\mathbf{Z}} = \mathcal{S}_\phi\left(\mathbf{Z}^T, \mathbf{X}, \mathbf{A}, \mathbf{M}\right), \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the target-domain relational graph and $\mathbf{M} \in \mathbb{R}^{N\times C_m}$ denotes optional node attributes. For forecasting, $\widetilde{\mathbf{Z}}$ is decoded by the *original frozen forecasting head* $\mathcal{D}_\Theta$:

$$\widehat{\mathbf{Y}} = \mathcal{D}_\Theta\left(\widetilde{\mathbf{Z}}\right), \qquad \nabla_\Theta \mathcal{L} = 0. \tag{3}$$

Thus, both the TSFM encoder and its native forecasting head remain frozen; only the graph adaptation parameters are optimized.

3.2 Relational Graph Token Alignment

The frozen TSFM does not explicitly encode target-domain graph structure. Graft therefore constructs relational graph tokens from historical observations, node attributes, and graph dependencies. Each scalar observation is projected into a hidden representation $\mathbf{H}^x$; neighborhood-aware histories are aggregated through $\mathbf{A}$, while node attributes are projected and broadcast along the temporal dimension. The self-history, neighborhood-history, and attribute representations are concatenated, temporally pooled, and propagated through two graph-convolutional layers with residual normalization. This produces temporally aligned relational graph tokens

$$\mathbf{Z}^G = \mathrm{Proj}(\mathrm{GraphAlign}(\mathbf{X}, \mathbf{A}, \mathbf{M})) \in \mathbb{R}^{B\times N\times D}, \tag{4}$$

which are expanded along the output-patch dimension to match $\mathbf{Z}^T$. The key design is that graph representations are conditioned on the observed temporal context rather than encoded independently. Detailed layer-wise equations are provided in Appendix B.

3.3 Temporal–Graph Fusion and Causal-Mediated Graph Adapter

Forecast and graph tokens are first projected into a shared latent space, $\mathbf{T} = \mathrm{MLP}_T(\mathbf{Z}^T)$ and $\mathbf{S} = \mathrm{MLP}_S(\mathbf{Z}^G)$. A dimension-wise gate

$$\mathbf{G} = \sigma(\mathrm{MLP}_G([\mathbf{T}; \mathbf{S}])) \tag{5}$$

controls graph injection, yielding

$$\mathbf{Z}^F = \mathrm{LN}(\mathbf{T} + \mathbf{G} \odot \mathbf{S}). \tag{6}$$

CMGA then models three complementary graph effects: direct local interactions on observed edges, mediated regional interactions through latent relational groups, and global contextual effects. A dynamic router balances these pathways, while a structural-strength gate controls the magnitude of the graph correction:

$$\mathbf{Z}^s = \alpha_d \mathbf{Z}^d + \alpha_m \mathbf{Z}^m + \alpha_g \mathbf{Z}^g, \qquad \widetilde{\mathbf{Z}} = \mathrm{LN}\big(\mathbf{Z}^F + \mathbf{C} \odot \mathbf{Z}^s\big). \tag{7}$$

The complete direct, mediated, global, routing, and modulation formulations are given in Appendix B.

3.4 Task-Specific Decoding

For spatio-temporal forecasting, the adapted tokens are passed through the original frozen TSFM forecasting head; no additional forecasting head is introduced. Only ϕ is optimized using the backbone-specific forecasting objective. To test cross-task generalization, we further instantiate Graft for multivariate time-series anomaly detection. Each variable becomes a graph node and the spatial graph is replaced by an inter-variable dependency graph $\mathbf{A}_{\mathrm{AD}}$. The same graph adaptation modules produce $\widetilde{\mathbf{Z}}^{\mathrm{AD}}$, followed by a lightweight *trainable reconstruction head* $\mathcal{D}_\psi^{\mathrm{rec}}$:

$$\widehat{\mathbf{X}}^{\mathrm{tar}} = \mathcal{D}_\psi^{\mathrm{rec}}\left(\widetilde{\mathbf{Z}}^{\mathrm{AD}}\right), \qquad \min_{\phi,\psi} \mathcal{L}_{\mathrm{rec}},\ \Theta \text{ fixed}. \tag{8}$$

The anomaly score is computed as the mean squared reconstruction error across variables. Full objectives are listed in Appendix B.

4 Experiments

4.1 Experimental Setup

We evaluate Graft on seven spatio-temporal forecasting benchmarks (PEMS03, PEMS04, PEMS07, PEMS08, PEMS-BAY, METR-LA, and NYC-BIKE) and four multivariate anomaly-detection benchmarks (SMD, SWaT, WADI, and PSM). Forecasting baselines cover time-series expert models, spatio-temporal expert models, TSFMs, and STFMs; anomaly-detection baselines include DAGMM, CAE-M, OmniAnomaly, USAD, GDN, MSCRED, TranAD, MAD-GAN, TimesNet, CARLA, and CATCH. Dataset statistics, baseline citations, frozen-backbone specifications, and implementation details are provided in Appendix C. We instantiate Graft with frozen Chronos-2 Ansari et al. (2025) and Moirai-1.1-R-small Woo et al. (2024) backbones.

4.2 Few-Shot Spatio-Temporal Forecasting

Setups. We evaluate the proposed Graft under a few-shot setting to examine whether frozen TSFMs can be efficiently adapted to downstream spatio-temporal forecasting tasks with limited supervision. Specifically, only the last 10% of the labeled training windows are used to optimize the lightweight graph adaptation modules, while the validation and test splits remain unchanged. Following common evaluation protocols Shao et al. (2022b;c), we consider both short-term forecasting with $12 \rightarrow 12$ and long-term forecasting with $96 \rightarrow 96$, to assess the adaptation ability of different models under different prediction horizons. To verify the generality of the proposed framework, we instantiate Graft with two different frozen temporal foundation backbones, namely Chronos-2 and Moirai-small, denoted as $\text{Graft}_{\text{Chronos}}$ and $\text{Graft}_{\text{Moirai}}$, respectively. For both variants, the parameters of the TSFM are kept completely frozen during graph adaptation. All compared methods are evaluated using the same training, validation, and test splits, and the same 10% labeled subset is used for few-shot adaptation.

Results. The few-shot results in Table 1 demonstrate that Graft can effectively enhance the downstream spatio-temporal forecasting capability of pre-trained TSFMs. Although the frozen TSFM backbones already contain general temporal forecasting priors, their raw temporal modeling ability is not sufficient to fully address graph-structured spatio-temporal forecasting, where spatial interactions among nodes play a critical role. By introducing graph adaptation, Graft consistently improves the forecasting performance of frozen TSFMs under limited supervision.

Table 1: Few-shot forecasting results under short-term and long-term prediction settings. Bold red indicates the best performance, while underlined blue denotes the second-best performance.

Type	Model	PEMS03		PEMS04		PEMS07		PEMS08		PEMS-BAY		METR-LA		NYC-BIKE	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Short-term Forecasting (12 → 12)															
TSEMs	Informer	23.24	37.98	29.81	45.59	37.55	62.55	31.69	51.53	2.96	6.23	4.93	9.20	2.02	7.32
	FEDFormer	19.66	30.15	25.22	38.46	29.41	43.96	20.49	31.84	2.58	5.37	5.31	10.55	1.96	7.13
	DLinear	21.94	35.30	28.37	44.57	31.89	49.65	23.10	36.35	2.21	5.20	4.57	9.82	2.87	8.64
	PatchTST	21.97	35.59	28.11	44.13	31.19	48.91	22.42	35.64	2.15	5.23	4.34	9.75	1.45	6.57
	TimeMixer	21.41	33.57	27.37	42.16	30.31	46.36	22.05	34.09	2.11	4.93	4.23	9.20	1.42	6.29
STEMs	STID	22.93	34.10	26.72	40.31	31.64	46.72	23.17	34.09	2.00	4.57	4.00	8.20	1.35	6.26
	STAEformer	30.79	47.67	48.23	68.46	33.50	51.43	36.15	51.05	2.01	4.62	4.61	8.91	1.53	7.63
	BigST	18.41	28.45	23.97	36.88	25.72	39.72	19.40	29.96	1.91	4.26	**3.72**	**7.19**	1.07	**4.74**
TSFMs	TimesFM	21.99	35.31	27.84	43.15	32.61	50.20	22.06	33.87	2.25	5.49	5.56	12.87	2.10	7.89
	Moirai	21.40	32.38	33.73	54.09	35.69	51.36	38.01	53.05	2.26	5.49	4.95	12.75	2.53	8.63
	Rose	23.00	36.21	30.85	46.17	35.81	53.73	24.79	37.44	2.19	5.14	4.84	12.53	1.99	7.52
STFMs	UniST	40.39	53.44	42.76	59.07	40.77	54.86	35.70	46.74	5.14	8.28	8.79	14.34	2.48	8.06
	OpenCity	17.90	28.80	24.78	40.41	44.43	65.47	32.16	48.47	2.77	6.08	4.18	8.33	2.35	7.21
	FactoST-v2	16.75	27.20	22.61	35.95	24.70	40.79	17.65	28.44	1.85	4.90	4.44	12.16	1.28	6.32
Ours	$\text{Graft}_{\text{Chronos}}$	**16.66**	**26.47**	**22.49**	**35.24**	**23.49**	37.57	17.81	28.23	**1.82**	4.34	5.12	12.66	0.96	5.28
	$\text{Graft}_{\text{Moirai}}$	16.82	26.61	22.86	36.13	23.85	**37.36**	**17.44**	**27.42**	1.87	**4.19**	5.78	12.23	**0.95**	5.09
Long-term Forecasting (96 → 96)															
TSEMs	Informer	46.27	69.41	54.26	83.42	52.82	81.78	44.25	68.43	3.27	6.81	6.38	12.29	2.84	9.39
	FEDFormer	71.02	113.29	93.52	124.07	74.74	108.02	108.39	143.49	4.20	7.34	9.71	16.09	2.80	9.37
	DLinear	76.41	113.63	85.61	125.44	106.68	147.43	76.77	109.15	4.62	9.52	7.65	13.42	3.21	10.97
	PatchTST	61.22	100.33	70.71	104.00	80.09	118.54	57.31	87.33	4.32	9.22	7.20	14.17	2.07	8.36
	TimeMixer	47.86	71.52	58.44	86.67	67.75	100.45	45.10	65.53	4.11	8.94	7.00	13.33	2.02	7.93
STEMs	STID	45.45	65.35	78.13	111.12	71.32	106.95	75.87	103.15	3.10	6.63	**5.94**	11.91	1.71	7.19
	STAEformer	77.42	115.67	64.12	91.95	61.45	91.06	68.45	96.14	3.28	6.65	6.15	**11.56**	2.18	8.99
	BigST	51.87	75.56	52.37	80.23	54.92	82.12	58.68	86.76	2.93	6.20	6.69	12.22	1.82	6.27
TSFMs	TimesFM	38.47	59.77	64.43	93.44	157.10	208.36	89.93	125.27	5.18	9.97	14.23	22.56	4.09	11.73
	Moirai	51.40	79.47	81.30	116.26	134.46	200.30	68.73	97.89	5.78	10.97	12.17	22.39	5.51	13.64
	Rose	76.67	108.90	100.22	137.06	137.43	181.66	82.34	112.47	4.55	9.18	8.49	21.24	5.46	13.05
STFMs	UniST	67.70	94.00	85.14	112.11	101.20	134.98	73.81	96.45	5.17	8.27	13.16	19.96	3.31	10.22
	OpenCity	34.21	54.82	67.24	112.20	50.70	78.36	49.47	82.07	7.40	12.38	9.71	13.62	4.66	12.78
	FactoST-v2	28.41	46.61	39.73	61.54	43.55	72.46	38.48	64.39	**2.59**	5.99	6.39	20.81	1.44	8.72
Ours	$\text{Graft}_{\text{Chronos}}$	**26.85**	**43.55**	**35.45**	**54.37**	**34.78**	**58.69**	**30.94**	**48.83**	2.86	6.21	10.69	20.10	**0.82**	**4.32**
	$\text{Graft}_{\text{Moirai}}$	27.87	43.57	35.85	54.82	37.44	58.84	33.20	52.03	2.81	**5.94**	10.86	20.62	0.95	5.01

Compared with dedicated spatio-temporal foundation models, Graft provides a more cost-effective adaptation paradigm. Existing STFMs typically require large-scale spatio-temporal pre-training and full-model fine-tuning on the target domain, which can be computationally expensive and difficult to generalize across heterogeneous spatial domains. In contrast, Graft keeps the temporal foundation backbone completely frozen and only optimizes lightweight graph adaptation modules on the target data. Despite this substantially lower cost, Graft achieves stronger or more competitive forecasting accuracy than existing STFMs in both short-term and long-term settings. This indicates that reusing a general time-series foundation model and learning a compact spatial adapter can be a more efficient and effective alternative to building a dedicated spatio-temporal foundation model from scratch.

The results also validate the generality of the proposed framework. Graft achieves consistent improvements when instantiated with different frozen TSFM backbones, including Chronos-2 and Moirai-small. This suggests that Graft is not tied to a specific temporal foundation model architecture and can integrate lightweight graph adaptation into diverse frozen TSFMs for downstream spatio-temporal forecasting tasks.

4.3 Zero-Shot Motivation and Adaptation Analysis

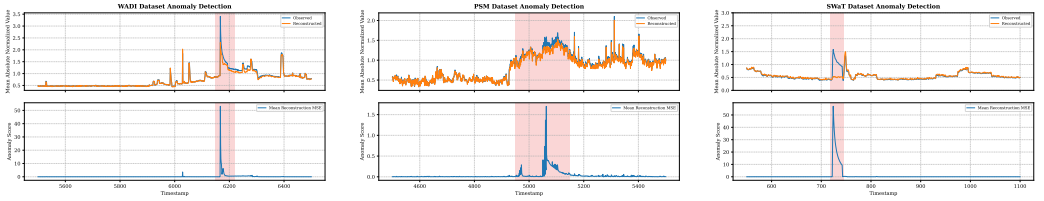
Before adaptation, we evaluate the raw frozen TSFM backbone against existing foundation models. The complete zero-shot comparison in Appendix D shows that pre-trained STFMs do not consistently outperform strong TSFMs across heterogeneous target graphs, supporting the separation of transferable temporal priors from target-specific graph structure. After 10% few-shot adaptation, Graft substantially reduces MAE and RMSE relative to its zero-shot backbone, with particularly large improvements at $96 \rightarrow 96$ (Appendix E). This directly motivates the Graft paradigm: preserve general temporal priors and incorporate graph structure only through lightweight target-domain adaptation modules.

Table 2: Anomaly detection performance comparison on SMD, SWaT, WADI, and PSM.

Dataset	**SMD**			**SWaT**			**WADI**			**PSM**		
Metric	$F1_T$	F1	$F1_{PA}$	$F1_T$	F1	$F1_{PA}$	$F1_T$	F1	$F1_{PA}$	$F1_T$	F1	$F1_{PA}$
DAGMM	0.379	0.435	0.840	0.370	0.770	0.841	0.406	0.279	0.363	0.458	0.478	0.899
CAE-M	0.332	0.397	0.833	0.288	0.723	0.812	0.238	0.138	0.208	0.444	0.447	0.823
USAD	0.364	0.426	0.841	0.386	0.770	0.824	0.406	0.279	0.375	0.451	0.432	0.841
OmniAnomaly	0.353	0.415	0.804	0.376	0.775	0.838	0.410	0.281	0.387	0.463	0.489	0.831
GDN	0.527	0.506	0.929	0.384	0.784	0.846	0.412	0.312	0.662	0.441	0.444	0.910
MSCRED	0.329	0.401	0.832	0.277	0.723	0.812	0.311	0.221	0.261	0.440	0.470	0.770
TranAD	0.390	0.457	0.827	0.376	0.782	0.846	0.353	0.328	0.671	0.516	**0.545**	0.863
MAD-GAN	0.352	0.419	0.839	0.370	0.773	0.833	0.353	0.259	0.275	0.475	0.486	0.895
TimesNet	0.374	0.442	0.851	0.387	0.774	0.807	0.382	0.291	0.448	0.452	0.441	0.846
CARLA	0.523	0.511	0.904	0.403	0.779	0.816	0.412	0.337	0.575	0.489	0.524	0.874
CATCH	0.517	0.529	0.931	0.396	0.781	0.850	0.419	0.351	0.675	0.501	0.473	0.914
Graft (Ours)	**0.585**	**0.554**	**0.935**	**0.551**	**0.799**	**0.862**	**0.521**	**0.466**	**0.766**	**0.571**	0.516	**0.952**

4.4 Cross-Task Generalization to Anomaly Detection

Setups. To evaluate the generalization ability of the proposed graph adaptation mechanism beyond spatio-temporal forecasting, we further conduct experiments on four widely used multivariate time-series anomaly detection benchmarks using Chronos-2 as the frozen TSFM backbone, including SMD, SWaT, WADI, and PSM. In contrast to spatio-temporal forecasting, where graph structures describe spatial dependencies among sensors or regions, each variable is treated as a graph node in anomaly detection, and the physical spatial topology is replaced by an inter-variable dependency graph. Following graph-based multivariate time-series anomaly detection methods Deng et al. (2021)–Qiao et al. (2026), we construct an inter-variable dependency graph using only normal training data. Specifically, instantaneous dependencies are estimated using absolute Pearson correlations, while lag-aware dependencies are measured using the maximum absolute cross-correlation within three temporal lags. The two dependency components are equally weighted, and the top-5 dependencies of each variable are retained to obtain a sparse relational graph. Following existing anomaly detection studies, we report $F1_T$, F1, and $F1_{PA}$ as the evaluation metrics. F1 denotes the standard point-wise F1 score. $F1_T$ is the range-based F1 score that considers anomaly intervals. $F1_{PA}$ represents the point-adjusted F1 score, where a predicted anomaly point within a ground-truth anomaly range indicates that the entire range is successfully detected. For all methods, the anomaly detection threshold is selected by searching for the optimal threshold on the test anomaly scores, following the same evaluation protocol to ensure a fair comparison.



(a) WADI (b) PSM (c) SWaT

Figure 3: Visualization of anomaly detection results on WADI, PSM, and SWaT. The upper panels compare the mean absolute normalized observations and reconstructions across variables, while the lower panels show the corresponding mean reconstruction MSE.

Results. Table 2 reports the anomaly detection results on four multivariate time-series benchmarks. Graft achieves the best performance on 11 out of 12 evaluation metrics, demonstrating that the proposed graph adaptation mechanism remains effective when the underlying graph semantics and downstream objective are both changed. In particular, Graft consistently improves all three metrics on SMD, SWaT, and WADI, and achieves the best $F1_T$ and $F1_{PA}$ on PSM. The gains are especially pronounced on SWaT and WADI, indicating the effectiveness of Graft in modeling abnormal behaviors under inter-variable dependency structures. Although Graft is primarily developed for spatio-temporal forecasting, these results show that its effectiveness is not tied to physical spatial topology. By replacing spatial graphs with inter-variable dependency graphs and forecasting with

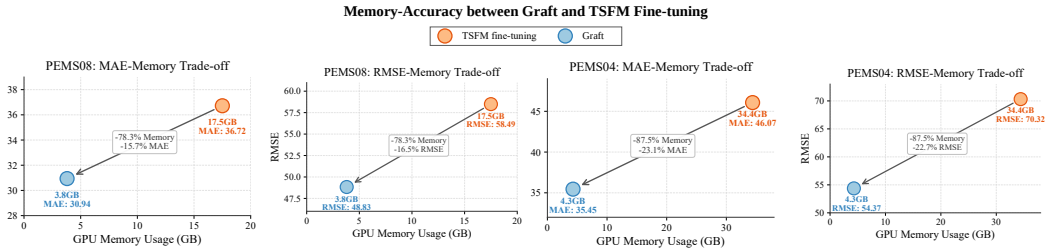


Figure 4: Memory-accuracy trade-off between Graft adaptation and full TSFM fine-tuning under the 10% few-shot setting, taking Chronos-2 as an example.

reconstruction-based detection, the same graph adaptation mechanism can be transferred to a different temporal-graph task without modifying the frozen TSFM backbone. This provides strong evidence for the cross-task generalizability of Graft across heterogeneous graph semantics and downstream objectives.

Figure 3 further visualizes the reconstruction behavior on representative anomaly intervals. The reconstructed signals closely follow the observed patterns during normal periods, while larger reconstruction deviations and increased anomaly scores appear around anomalous intervals. These results qualitatively support the effectiveness of Graft for anomaly detection under inter-variable graph structures.

4.5 Efficiency Analysis

Fig. 4 compares the memory-accuracy trade-off between Graft and direct TSFM fine-tuning in the 10% few-shot and 96-to-96 forecasting settings (using the Chronos-2 foundation model as an example). The results show that Graft training consistently achieves higher forecasting accuracy while substantially reducing GPU memory consumption across both datasets. On PEMS08, the memory usage is reduced from 17.5 GB to 3.8 GB, corresponding to a 78.3% reduction, while MAE and RMSE are further improved by 15.7% and 16.5%, respectively. The advantage becomes even more pronounced on PEMS04, where the memory consumption decreases from 34.4 GB to 4.3 GB, yielding an 87.5% reduction, together with 23.1% and 22.7% improvements in MAE and RMSE. These results indicate that directly fine-tuning TSFM under limited downstream data is not only computationally expensive but also less effective for spatio-temporal forecasting. In contrast, freezing the TSFM backbone and optimizing only the lightweight graph adaptation modules provides a more favorable efficiency-effectiveness trade-off.

Fig. 5 presents the inference time decomposition of Graft with different frozen TSFM backbones on PEMS03. All inference times are reported as the wall-clock latency for one complete test-set evaluation pass. The results show that the overall inference time of Graft is primarily determined by the selected frozen TSFM backbone. When the backbone changes from Chronos-2 to Moirai-small, the full inference time varies substantially, while the combined adaptation and forecast-head inference time remains relatively small. This indicates that the combined cost of graph adaptation and forecast decoding remains small relative to the overall inference cost of the frozen backbone. Therefore, the proposed framework can effectively adapt frozen TSFMs to spatio-temporal forecasting while preserving the deployment efficiency of the selected backbone.

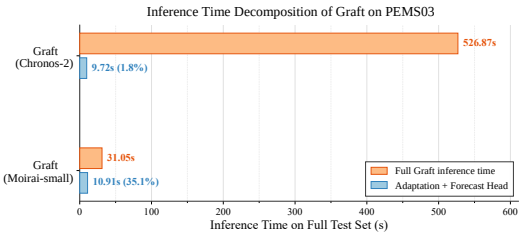


Figure 5: Inference time decomposition of Graft with different frozen TSFM backbones on PEMS03.

4.6 Comparison of Fine-Tuning Strategies

Table 3 compares Graft with representative fine-tuning strategies under the 10% few-shot and 96-to-96 forecasting setting. Graft consistently achieves the best performance across all four datasets.

Table 3: Comparison of different fine-tuning strategies under the 10% few-shot and 96-to-96 forecasting setting.

Dataset	Metric	Head-only	LoRA	Full Fine-tuning	Graft
PEMS08	MAE	46.47	41.81	36.72	**30.94**
	RMSE	68.93	65.67	58.49	**48.83**
PEMS03	MAE	40.71	36.98	31.23	**26.85**
	RMSE	62.40	60.52	52.46	**43.55**
PEMS04	MAE	57.19	53.83	46.07	**35.45**
	RMSE	84.48	80.77	70.32	**54.37**
PEMS-BAY	MAE	3.95	3.64	3.39	**2.86**
	RMSE	8.10	7.89	6.97	**6.21**

Table 4: Ablation study results of Graft under the 96-to-96 forecasting setting.

Model Variant	PEMS08		PEMS03		PEMS04		PEMS-BAY	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Graft	**30.94**	**48.83**	**26.85**	**43.55**	**35.45**	**54.37**	**2.86**	**6.21**
w/ Attn.-adapter	33.05	49.59	30.69	49.89	39.25	61.42	3.86	8.07
w/o CMGA	34.18	52.22	27.97	45.03	38.58	58.66	3.27	6.92
w/o graph align. & fus.	35.54	55.57	34.76	56.46	50.06	75.56	3.35	7.32

Although LoRA Hu et al. (2022) and full fine-tuning improve over head-only adaptation by updating the temporal representations of the TSFM, they do not explicitly exploit target-domain spatial structure. In contrast, Graft directly incorporates graph dependencies into the latent forecast representations while keeping the TSFM backbone frozen, leading to consistently better forecasting accuracy with lightweight adaptation.

4.7 Ablation studies

Table 4 reports the ablation results under the 10% few-shot and 96-to-96 forecasting setting. The complete Graft consistently achieves the best performance across all datasets. Replacing or removing CMGA degrades forecasting accuracy, confirming the effectiveness of structured graph refinement beyond generic attention. The largest performance drop occurs when graph token alignment and fusion are removed; for example, the MAE increases from 35.45 to 50.06 on PEMS04 and from 26.85 to 34.76 on PEMS03. These results demonstrate that both relational graph token construction and CMGA are important for effectively integrating target-domain graph structure with the temporal priors of frozen TSFMs.

5 Conclusion

We presented Graft, a framework that integrates lightweight graph adaptation into frozen TSFMs for downstream spatio-temporal forecasting. By keeping the TSFM encoder and its native forecasting head frozen, Graft reuses transferable temporal priors while learning lightweight target-domain graph adaptation with limited supervision. Experiments on seven spatio-temporal forecasting benchmarks show strong few-shot performance across different frozen backbones, together with favorable memory–accuracy trade-offs relative to direct TSFM fine-tuning. Additional multivariate anomaly-detection experiments demonstrate that the graph adaptation mechanism generalizes across different graph semantics and downstream objectives. These results support lightweight graph adaptation as an effective way to extend frozen TSFMs to graph-structured temporal tasks.

AI Use Statement

Generative AI tools were used solely to polish the manuscript and improve its readability. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content of the paper.

Reproducibility Statement

Appendix C reports dataset statistics, backbone specifications, implementation details, and experimental protocols; Appendix D provides the complete zero-shot forecasting comparison; and Appendix E contains additional few-shot sensitivity, zero-to-few-shot adaptation, node-wise visualization, cross-task graph adaptation, and random-seed robustness analyses. The source code and complete experiment configurations will be publicly released upon acceptance. If requested during the review process, an anonymized version of the code will be made available to the reviewers and area chairs.

References

Li Y, Yu R, Shahabi C, et al. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting[C]//*International Conference on Learning Representations*. 2018.

Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2017, 31(1).

Liang Y, Wen H, Xia Y, et al. Foundation models for spatio-temporal data science: A tutorial and survey[C]//*Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 2025: 6063-6073.

Liang Y, Wen H, Nie Y, et al. Foundation models for time series analysis: A tutorial and survey[C]//*Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 2024: 6555-6565.

Yuan Y, Ding J, Feng J, et al. Unist: A prompt-empowered universal model for urban spatio-temporal prediction[C]//*Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 2024: 4095-4106.

Li Z, Xia L, Shi L, et al. Opencity: Open spatio-temporal foundation models for traffic prediction[C]//*in Proceedings of the ACM Web Conference*. 2025.

Zhong S, Qiu J, Wu Y, et al. Learning to factorize spatio-temporal foundation models[C]//*Advances in Neural Information Processing Systems*. 2025, 38: 98047-98075.

Zhong S, Qiu J, Wu Y, et al. Learning to Factorize and Adapt: A Versatile Approach Toward Universal Spatio-Temporal Foundation Models[J]. *arXiv preprint arXiv:2601.12083*, 2026.

Ansari A F, Stella L, Turkmen A C, et al. Chronos: Learning the Language of Time Series[J]. *Transactions on Machine Learning Research*, 2024.

Das A, Kong W, Sen R, et al. A decoder-only foundation model for time-series forecasting[C]//*Proceedings of the 41st International Conference on Machine Learning*. 2024: 10148-10167.

Rasul K, Ashok A, Williams A R, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting[J]. *arXiv preprint arXiv:2310.08278*, 2023.

Woo G, Liu C, Kumar A, et al. Unified training of universal time series forecasting transformers[C]// *Forty-first International Conference on Machine Learning*. 2024.

Goswami M, Szafer K, Choudhry A, et al. MOMENT: A Family of Open Time-series Foundation Models[C]//*Forty-first International Conference on Machine Learning*. 2024.

Liu Y, Zhang H, Li C, et al. Timer: generative pre-trained transformers are large time series models[C]//*Proceedings of the 41st International Conference on Machine Learning*. 2024.

Jin M, Wang S, Ma L, et al. Time-llm: Time series forecasting by reprogramming large language models[C]//*International conference on learning representations*. 2024.

Houlsby N, Giurgiu A, Jastrzebski S, et al. Parameter-efficient transfer learning for NLP[C]//*International conference on machine learning*. 2019: 2790-2799.

Hu E J, Wallis P, Allen-Zhu Z, et al. LoRA: Low-Rank Adaptation of Large Language Models[C]//*International Conference on Learning Representations*. 2022.

Kipf T N, Welling M. Semi-Supervised Classification with Graph Convolutional Networks[C]//*International Conference on Learning Representations*. 2017.

Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting[C]//*Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018: 3634-3640.

Wu Z, Pan S, Long G, et al. Graph wavenet for deep spatial-temporal graph modeling[C]//*Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2019: 1907-1913.

Bai L, Yao L, Li C, et al. Adaptive graph convolutional recurrent network for traffic forecasting[J]. *Advances in neural information processing systems*, 2020, 33: 17804-17815.

Wu Z, Pan S, Long G, et al. Connecting the dots: Multivariate time series forecasting with graph neural networks[C]//*Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020: 753-763.

Jiang J, Han C, Zhao W X, et al. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2023, 37(4): 4365-4373.

Liu H, Dong Z, Jiang R, et al. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting[C]//*Proceedings of the 32nd ACM international conference on information and knowledge management*. 2023: 4125-4129.

Shao Z, Zhang Z, Wang F, et al. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting[C]//*Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2022: 1567-1577.

Guo S, Lin Y, Wan H, et al. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2021, 34(11): 5415-5428.

Li Z, Xia L, Tang J, et al. Urbangpt: Spatio-temporal large language models[C]//*Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 2024: 5351-5362.

Ansari A F, Shchur O, Küken J, et al. Chronos-2: From univariate to universal forecasting[J]. *arXiv preprint* arXiv:2510.15821, 2025.

Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2021, 35(12): 11106-11115.

Zhou T, Ma Z, Wen Q, et al. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting[C]//*International conference on machine learning. PMLR*, 2022: 27268-27286.

Zeng A, Chen M, Zhang L, et al. Are transformers effective for time series forecasting?[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2023, 37(9): 11121-11128.

Nie Y, Nguyen N H, Sinthong P, et al. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers[C]//*International Conference on Learning Representations*. 2023.

Wang S, Wu H, Shi X, et al. Timemixer: Decomposable multiscale mixing for time series forecasting[C]//*International conference on learning representations*. 2024.

Shao Z, Zhang Z, Wang F, et al. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting[C]//*Proceedings of the 31st ACM international conference on information & knowledge management*. 2022: 4454-4458.

Han J, Zhang W, Liu H, et al. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks[J]. *Proceedings of the VLDB Endowment*, 2024, 17(5): 1081-1090.

Wang Y, Qiu Y, Chen P, et al. Towards a General Time Series Forecasting Model with Unified Representation and Adaptive Transfer[C]//*International Conference on Machine Learning*. 2025.

Shao Z, Zhang Z, Wei W, et al. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting[J]. *Proceedings of the VLDB Endowment*, 2022, 15(11): 2733-2746.

Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, 35(5): 4027–4035.

Xu K, Li Y, Li Y, et al. Masked graph neural networks for unsupervised anomaly detection in multivariate time series[J]. *Sensors*, 2023, 23(17): 7552.

Qiao C, Fu X, Chen Z, et al. Multi-entity time series anomaly detection method based on graph attention and cross-correlation analysis[J]. *Expert Systems with Applications*, 2026, 307: 130996.

Su Y, Zhao Y, Niu C, et al. Robust anomaly detection for multivariate time series through stochastic recurrent neural network[C]//*Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019: 2828–2837.

Goh J, Adepu S, Junejo K N, et al. A dataset to support research in the design of secure water treatment systems[C]//*Critical Information Infrastructures Security*, 2017: 88–99.

Ahmed C M, Palleti V R, Mathur A P. WADI: A water distribution testbed for research in the design of secure cyber physical systems[C]//*Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 2017: 25–28.

Abdulaal A, Liu Z, Lancewicki T. Practical approach to asynchronous multivariate time series anomaly detection and localization[C]//*Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021: 2485–2494.

Zong B, Song Q, Min M R, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection[C]//*International conference on learning representations*. 2018.

Zhang Y, Chen Y, Wang J, et al. Unsupervised deep anomaly detection for multi-sensor time-series signals[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2021, 35(2): 2118-2132.

Audibert J, Michiardi P, Guyard F, et al. Usad: Unsupervised anomaly detection on multivariate time series[C]//*Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020: 3395-3404.

Zhang C, Song D, Chen Y, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2019, 33(01): 1409-1416.

Tuli S, Casale G, Jennings N R. Tranad: Deep transformer networks for anomaly detection in multivariate time series data[J]. *arXiv preprint arXiv*:2201.07284, 2022.

Li D, Chen D, Jin B, et al. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks[C]//*International conference on artificial neural networks. Cham: Springer International Publishing*. 2019: 703-716.

Wu H, Hu T, Liu Y, et al. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis[C]//*International Conference on Learning Representations*. 2023.

Darban Z Z, Webb G I, Pan S, et al. CARLA: Self-supervised contrastive representation learning for time series anomaly detection[J]. *Pattern Recognition*, 2025, 157: 110874.

Wu X, Qiu X, Li Z, et al. CATCH: Channel-Aware Multivariate Time Series Anomaly Detection via Frequency Patching[C]//*The Thirteenth International Conference on Learning Representations*. 2025.

A Problem Formulation and Notation

Spatio-Temporal Data. We formulate spatio-temporal forecasting as modeling the evolution of dynamic signals over a set of spatially related entities. Let $\mathcal{D} = \{(\mathbf{X}, \mathbf{A}, \mathbf{M})\}$ denote a spatio-temporal dataset, where $\mathbf{X}$ represents temporal observations, $\mathbf{A}$ describes graph-structured dependencies among nodes, and $\mathbf{M}$ denotes optional node attributes. Specifically, let $\mathcal{V} = \{v_1, \ldots, v_N\}$ be a set of N spatial entities, such as sensors, regions, or stations. At each time step t, the system state is denoted as

$$\mathbf{X}_t \in \mathbb{R}^{N \times C}, \tag{9}$$

where C is the number of observed variables. Given a historical window of length L, the input sequence is represented as

$$\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{N \times L \times C}. \tag{10}$$

Let $\mathbf{M} \in \mathbb{R}^{N \times C_m}$ denote optional node attributes, where C_m is the attribute dimension.

Graph-Structured Dependencies. The graph-structured dependencies among entities are represented by an adjacency matrix

$$\mathbf{A} \in \mathbb{R}^{N \times N}, \tag{11}$$

where $\mathbf{A}_{ij}$ characterizes the relational dependency between nodes v_i and v_j. Depending on the application scenario, $\mathbf{A}$ can be derived from physical connectivity, geographical distance, functional similarity, or data-driven correlations. Since different spatio-temporal domains often involve different node sets, spatial layouts, and interaction mechanisms, graph-structured dependency modeling is usually highly domain-specific.

Spatio-Temporal Forecasting. Given historical observations $\mathbf{X}_{t-L+1:t}$ and graph-structured dependency information $\mathbf{A}$, the goal of spatio-temporal forecasting is to predict the future sequence of length H:

$$\mathbf{Y}_{t+1:t+H} \in \mathbb{R}^{N \times H \times C_y}, \tag{12}$$

where C_y is the number of target variables. The forecasting task can be generally formulated as learning a mapping function f_θ:

$$\hat{\mathbf{Y}}_{t+1:t+H} = f_\theta\left(\mathbf{X}_{t-L+1:t}, \mathbf{A}, \mathbf{M}\right), \tag{13}$$

where $\hat{\mathbf{Y}}_{t+1:t+H}$ denotes the predicted future sequence.

Pre-trained Time-Series Foundation Models. A pre-trained time-series foundation model aims to learn transferable temporal priors from large-scale time-series corpora. Given a historical sequence $\mathbf{x}_{t-L+1:t}$, a temporal foundation model parameterized by Θ can be written as

$$\hat{\mathbf{y}}_{t+1:t+H} = \mathcal{F}_\Theta\left(\mathbf{x}_{t-L+1:t}\right). \tag{14}$$

Although such models provide strong general temporal forecasting ability, they are usually designed for univariate or multivariate time-series forecasting and do not natively incorporate explicit target-domain graph structures. For spatio-temporal forecasting, such graph structures primarily describe spatial dependencies among sensors, regions, or nodes.

B Detailed Architecture and Objectives

B.1 Frozen TSFM Forecast Token Extraction

Given the historical observations $\mathbf{X} \in \mathbb{R}^{B \times N \times L}$, Graft first reshapes them into BN node-wise time series:

$$\mathbf{X}^{\text{flat}} = \text{Reshape}(\mathbf{X}) \in \mathbb{R}^{BN \times L}. \tag{15}$$

These node-wise sequences are encoded by the frozen TSFM encoder. Since the output horizon H is represented by P output patches, we extract the last P hidden states as future-oriented forecast tokens:

$$\mathbf{Z}^T_{\text{flat}} = \mathcal{E}_\Theta\left(\mathbf{X}^{\text{flat}}\right)_{[-P:]} \in \mathbb{R}^{BN \times P \times D}. \tag{16}$$

Then, the flattened tokens are reshaped back to the node-level form:

$$\mathbf{Z}^T = \text{Reshape}\left(\mathbf{Z}^T_{\text{flat}}\right) \in \mathbb{R}^{B \times N \times P \times D}. \tag{17}$$

These tokens serve as the latent forecasting representations of the frozen TSFM and provide transferable temporal priors for downstream graph adaptation.

B.2 Relational Graph Token Alignment

Although the frozen TSFM provides strong temporal forecasting priors, it does not explicitly encode target-domain graph structures. We therefore construct relational graph tokens by jointly modeling historical observations, node attributes, and graph-structured dependencies. Here, temporal–graph alignment is performed during graph token construction: graph-structured dependency representations are conditioned on the observed temporal context rather than encoded independently from the historical signals.

First, each scalar observation is projected into a temporal hidden representation:

$$\mathbf{H}^x = \mathrm{MLP}_x\left(\mathbf{X}\right) \in \mathbb{R}^{B\times N\times L\times d_h}. \tag{18}$$

The neighborhood-aware historical representation is obtained by aggregating node embeddings according to the relational graph:

$$\mathbf{H}^n_{:,i,:,:} = \sum_{j=1}^{N} \mathbf{A}_{ij}\mathbf{H}^x_{:,j,:,:}. \tag{19}$$

Meanwhile, node attributes are projected into the same hidden dimension:

$$\mathbf{H}^m = \mathrm{MLP}_m\left(\mathbf{M}\right) \in \mathbb{R}^{N\times d_h}. \tag{20}$$

The node attribute embeddings are broadcast along the batch and temporal dimensions and concatenated with the self-history and neighborhood-history representations:

$$\mathbf{H}^c = \mathrm{MLP}_c\left(\left[\mathbf{H}^x;\mathbf{H}^n;\mathbf{H}^m\right]\right). \tag{21}$$

We then perform temporal pooling to obtain node-level graph-context representations:

$$\mathbf{H}^0 = \mathrm{MeanPool}_L\left(\mathbf{H}^c\right) \in \mathbb{R}^{B\times N\times d_h}. \tag{22}$$

To further propagate relational information over the graph, we employ two GCN layers with residual connections Kipf et al. (2017). We first add self-loops to the relational graph:

$$\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}, \tag{23}$$

where $\mathbf{I}$ denotes the identity matrix. The degree matrix and symmetrically normalized adjacency matrix are defined as

$$\widetilde{\mathbf{D}}_{ii} = \sum_{j=1}^{N} \widetilde{\mathbf{A}}_{ij}, \qquad \widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}. \tag{24}$$

The graph convolution at layer ℓ is then written as

$$\mathrm{GCN}_\ell\left(\mathbf{H},\mathbf{A}\right) = \widehat{\mathbf{A}}\mathbf{H}\mathbf{W}_\ell, \tag{25}$$

where $\mathbf{W}_\ell$ denotes the learnable transformation matrix of the ℓ-th GCN layer. The graph-context representations are updated by

$$\mathbf{H}^1 = \mathrm{LN}\left(\mathbf{H}^0 + \sigma\left(\mathrm{GCN}_1\left(\mathbf{H}^0,\mathbf{A}\right)\right)\right), \tag{26}$$

$$\mathbf{H}^2 = \mathrm{LN}\left(\mathbf{H}^1 + \sigma\left(\mathrm{GCN}_2\left(\mathbf{H}^1,\mathbf{A}\right)\right)\right), \tag{27}$$

where $\sigma(\cdot)$ denotes the GELU activation and $\mathrm{LN}(\cdot)$ denotes layer normalization.

After an FFN block and output projection, we obtain temporally aligned relational graph tokens:

$$\mathbf{Z}^G = \mathrm{Proj}\left(\mathrm{LN}\left(\mathbf{H}^2 + \mathrm{FFN}\left(\mathbf{H}^2\right)\right)\right) \in \mathbb{R}^{B\times N\times D}. \tag{28}$$

Finally, $\mathbf{Z}^G$ is expanded along the output-patch dimension:

$$\mathbf{Z}^G \leftarrow \mathrm{Expand}\left(\mathbf{Z}^G\right) \in \mathbb{R}^{B\times N\times P\times D}. \tag{29}$$

B.3 TEMPORAL–GRAPH TOKEN FUSION

The relational graph tokens $\mathbf{Z}^G$ encode graph-structured dependencies conditioned on the observed temporal context, while the forecast tokens $\mathbf{Z}^T$ capture future-oriented temporal priors from the frozen TSFM. To integrate these complementary representations, we introduce a gated temporal–graph fusion mechanism in the latent forecast-token space.

Given forecast tokens $\mathbf{Z}^T$ and relational graph tokens $\mathbf{Z}^G$, we first project them into a shared latent space:

$$\mathbf{T} = \mathrm{MLP}_T\left(\mathbf{Z}^T\right), \mathbf{S} = \mathrm{MLP}_S\left(\mathbf{Z}^G\right). \tag{30}$$

A dimension-wise fusion gate is then computed as

$$\mathbf{G} = \sigma\left(\mathrm{MLP}_G\left([\mathbf{T}; \mathbf{S}]\right)\right), \tag{31}$$

where $\sigma(\cdot)$ denotes the sigmoid function. The fused tokens are obtained by

$$\mathbf{Z}^F = \mathrm{LN}\left(\mathbf{T} + \mathbf{G} \odot \mathbf{S}\right), \tag{32}$$

where $\odot$ denotes element-wise multiplication. This design enables the model to adaptively determine how much target-domain graph information should be incorporated into each node, patch, and hidden dimension while preserving the temporal forecasting priors encoded by the frozen TSFM.

B.4 CAUSAL-MEDIATED GRAPH ADAPTER

Although gated temporal–graph fusion introduces graph-structured information into the forecast-token space, relational dependencies in real-world systems are often heterogeneous and multi-level. Local neighboring nodes may exert direct effects, latent regions or relational groups may mediate indirect dependencies, and the overall system may provide global contextual effects. To model these complementary graph interactions, we propose the **Causal-Mediated Graph Adapter (CMGA)**.

Causal-mediation interpretation. CMGA is motivated by the structural decomposition in causal mediation analysis, where relational influence may be transmitted through both direct and mediated pathways. In our formulation, the direct branch models edge-level relational influence between neighboring nodes, while the mediated branch represents indirect influence transmitted through latent relational groups. The global branch further accounts for system-level contextual variation. Accordingly, CMGA decomposes graph-structured interactions into direct, mediated, and global pathways before adaptively recombining them through a dynamic router. We emphasize that this decomposition provides a causal-mediation-inspired structural inductive bias rather than formal causal-effect identification. The relational graph is not assumed to be an identified causal DAG, and the model does not rely on interventional or counterfactual supervision.

Given fused tokens $\mathbf{Z}^F \in \mathbb{R}^{B\times N\times P\times D}$, CMGA first incorporates node-specific structural semantics by projecting node attributes into the token space. Let $\mathbf{M} \in \mathbb{R}^{N\times C_m}$ denote auxiliary node descriptors, which are either provided by the dataset or constructed from node identities and graph structural statistics. A learnable projection matrix $\mathbf{W}_m \in \mathbb{R}^{C_m\times D}$ maps these node attributes into the token dimension:

$$\mathbf{E} = \mathbf{M}\mathbf{W}_m \in \mathbb{R}^{N\times D}. \tag{33}$$

The node-aware tokens are computed as

$$\mathbf{X}^g_{:,i,:,:} = \mathbf{Z}^F_{:,i,:,:} + \mathbf{E}_i, \tag{34}$$

where $\mathbf{X}^g \in \mathbb{R}^{B\times N\times P\times D}$.

B.4.1 DIRECT LOCAL EFFECT BRANCH

The direct branch captures local edge-level interactions over the observed relational graph. Instead of densely computing all pairwise interactions, CMGA only computes messages on existing graph edges. Let $\mathcal{E} = \{(i,j) \mid A_{ij} > 0\}$ denote the set of valid relational edges between nodes i and j. For each edge (i,j), we construct an edge representation by combining the source token, target token, their difference, and their element-wise interaction:

$$\mathbf{e}_{ij} = \left[\mathbf{X}^g_{:,i,:,:}; \mathbf{X}^g_{:,j,:,:}; \mathbf{X}^g_{:,i,:,:} - \mathbf{X}^g_{:,j,:,:}; \mathbf{X}^g_{:,i,:,:} \odot \mathbf{X}^g_{:,j,:,:}\right]. \tag{35}$$

A learnable edge gate is then estimated by

$$\boldsymbol{\gamma}_{ij} = \sigma\left(\mathrm{MLP}_e\left(\mathbf{e}_{ij}\right)\right), \tag{36}$$

where $\boldsymbol{\gamma}_{ij} \in \mathbb{R}^{B\times P\times 1}$ measures the adaptive contribution of edge (i,j) for each sample and output patch.

The gated neighbor representation of node j is computed as

$$\mathbf{N}_j = \frac{\sum_{i:(i,j)\in\mathcal{E}} \boldsymbol{\gamma}_{ij} \odot \mathbf{X}^g_{:,i,:,:}}{\sum_{i:(i,j)\in\mathcal{E}} \boldsymbol{\gamma}_{ij} + \epsilon}, \tag{37}$$

where ϵ is a small constant for numerical stability. The direct local effect is then obtained by combining the self token and the gated neighbor token:

$$\mathbf{U}^d = \mathbf{X}^g\mathbf{W}^d_s + \mathbf{N}\mathbf{W}^d_n, \widetilde{\mathbf{Z}}^d = \mathrm{LN}\left(\mathbf{X}^g + \sigma\left(\mathbf{U}^d\right)\right), \mathbf{Z}^d = \mathrm{LN}\left(\widetilde{\mathbf{Z}}^d + \mathrm{FFN}_d\left(\widetilde{\mathbf{Z}}^d\right)\right). \tag{38}$$

where $\mathbf{W}^d_s, \mathbf{W}^d_n \in \mathbb{R}^{D\times D}$ are learnable projection matrices for the self token and gated neighbor token, respectively. This branch models direct graph interactions transmitted through local edges.

B.4.2 Mediated Regional Effect Branch

Local interactions alone may be insufficient to capture indirect dependencies among nodes. For example, nodes that are not directly connected may still interact through latent regions or relational groups. Therefore, CMGA introduces a mediated branch to model indirect graph interactions through latent higher-level structures.

Given node attributes $\mathbf{M}$, we learn a soft assignment matrix:

$$\mathbf{R} = \mathrm{Softmax}\left(\mathbf{M}\mathbf{W}_r\right) \in \mathbb{R}^{N\times K}, \tag{39}$$

where K is the number of latent regions or relational groups. The column-normalized assignment matrix is denoted as

$$\bar{\mathbf{R}}_{ik} = \frac{\mathbf{R}_{ik}}{\sum_{i=1}^{N} \mathbf{R}_{ik} + \epsilon}. \tag{40}$$

Node tokens are first aggregated into mediator tokens:

$$\mathbf{Q} = \bar{\mathbf{R}}^\top \mathbf{X}^g \in \mathbb{R}^{B\times K\times P\times D}. \tag{41}$$

The mediator-level adjacency matrix is induced from the node-level graph:

$$\mathbf{A}^r = \bar{\mathbf{R}}^\top \mathbf{A} \bar{\mathbf{R}}. \tag{42}$$

After adding self-loops and applying symmetric normalization, we obtain $\widehat{\mathbf{A}}^r$. The mediator-level neighborhood representation is computed as

$$\mathbf{Q}^n = \widehat{\mathbf{A}}^r \mathbf{Q}. \tag{43}$$

Then, the mediator tokens are updated by

$$\mathbf{U}^m = \mathbf{Q}\mathbf{W}^m_s + \mathbf{Q}^n\mathbf{W}^m_n, \widetilde{\mathbf{Q}}^m = \mathrm{LN}\left(\mathbf{Q} + \sigma\left(\mathbf{U}^m\right)\right), \mathbf{Q}^m = \mathrm{LN}\left(\widetilde{\mathbf{Q}}^m + \mathrm{FFN}_m\left(\widetilde{\mathbf{Q}}^m\right)\right). \tag{44}$$

Finally, the mediated effect is projected back to the node level:

$$\mathbf{Z}^m = \mathbf{R}\left(\mathbf{Q}^m\mathbf{W}^m_b\right) \in \mathbb{R}^{B\times N\times P\times D}. \tag{45}$$

This branch captures indirect graph dependencies mediated by latent higher-level relational structures.

B.4.3 Global Contextual Effect Branch

In addition to local and mediated effects, graph-structured temporal systems may also be influenced by global states. CMGA therefore introduces a global contextual branch. The global token is obtained by averaging node tokens:

$$\mathbf{G}^c = \frac{1}{N}\sum_{i=1}^{N} \mathbf{X}^g_{:,i,:,:}. \tag{46}$$

The global token is then broadcast to all nodes and concatenated with node-level tokens. A global-context representation is obtained by

$$\mathbf{U}^g = \mathrm{MLP}_g\left([\mathbf{X}^g; \mathbf{G}^c]\right), \tag{47}$$

where $\mathbf{U}^g \in \mathbb{R}^{B\times N\times P\times D}$ denotes the fused representation of node-level tokens and the global contextual token. The global contextual effect is then computed as

$$\mathbf{Z}^g = \mathrm{LN}\left(\mathbf{U}^g + \mathrm{FFN}_g\left(\mathbf{U}^g\right)\right). \tag{48}$$

This branch provides system-level contextual information for each node and output patch.

B.4.4 Dynamic Routing and Structural-Strength Modulation

After obtaining the three graph effects $\mathbf{Z}^d$, $\mathbf{Z}^m$, and $\mathbf{Z}^g$, CMGA dynamically balances their contributions. The routing weights are computed as

$$\boldsymbol{\alpha} = \mathrm{Softmax}\left(\mathrm{MLP}_r\left(\left[\mathbf{X}^g; \mathbf{Z}^d; \mathbf{Z}^m; \mathbf{Z}^g\right]\right)\right) \in \mathbb{R}^{B\times N\times P\times 3}. \tag{49}$$

The routed graph representation is then

$$\mathbf{Z}^s = \alpha_d \mathbf{Z}^d + \alpha_m \mathbf{Z}^m + \alpha_g \mathbf{Z}^g. \tag{50}$$

Meanwhile, a structural-strength gate is used to estimate the injection intensity of graph-structured corrections:

$$\mathbf{C} = \sigma\left(\mathrm{MLP}_c\left(\left[\mathbf{X}^g; \mathbf{Z}^d; \mathbf{Z}^m\right]\right)\right) \in \mathbb{R}^{B\times N\times P\times D}. \tag{51}$$

The structural-strength gate controls how strongly the routed graph effect modifies the fused temporal representation. The final CMGA output is computed by residual injection:

$$\widetilde{\mathbf{Z}} = \mathrm{LN}\left(\mathbf{Z}^F + \mathbf{C}\odot\mathbf{Z}^s\right). \tag{52}$$

Here, the router determines the relative composition of direct local, mediated regional, and global contextual effects, while the structural-strength gate controls the overall injection magnitude of the graph-structured correction. This design enables CMGA to adaptively inject target-domain relational information while reducing the risk of over-imposing noisy or task-specific graph patterns on the frozen temporal foundation model.

B.5 Task-Specific Decoding and Objectives

B.5.1 Spatio-Temporal Forecasting

For the primary spatio-temporal forecasting task, the adapted forecast tokens $\widetilde{\mathbf{Z}} \in \mathbb{R}^{B\times N\times P\times D}$ are reshaped into the flattened form:

$$\widetilde{\mathbf{Z}}_{\mathrm{flat}} \in \mathbb{R}^{BN\times P\times D}. \tag{53}$$

They are then decoded by the original frozen forecasting head of the pre-trained TSFM:

$$\hat{\mathbf{Y}} = \mathcal{D}_\Theta\left(\widetilde{\mathbf{Z}}_{\mathrm{flat}}\right). \tag{54}$$

No additional forecasting head is introduced, and both the TSFM encoder and its native forecasting head remain frozen during adaptation. For probabilistic TSFMs that produce multiple quantile forecasts, the output can be written as

$$\hat{\mathbf{Y}} = \left\{\hat{\mathbf{Y}}^{(q)}\right\}_{q\in\mathcal{Q}}, \tag{55}$$

where $\mathcal{Q}$ denotes the set of quantile levels. The point forecast is obtained from the median quantile:

$$\hat{\mathbf{Y}}^{\mathrm{point}} = \hat{\mathbf{Y}}^{(0.5)}. \tag{56}$$

During training, the TSFM encoder and decoder parameters Θ remain frozen, and only the graph adaptation parameters ϕ are optimized. For Chronos-2, we retain its quantile forecasting objective,

while for Moirai-small, the decoded point forecasts are optimized using mean squared error. For probabilistic TSFMs with quantile outputs, such as Chronos-2, the forecasting loss is defined as

$$\mathcal{L}_{\text{fore}} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \rho_q \left(\mathbf{Y} - \hat{\mathbf{Y}}^{(q)} \right), \tag{57}$$

where

$$\rho_q(u) = \max\left(qu, (q-1)u\right). \tag{58}$$

For Chronos-2, the forecasting optimization objective is

$$\min_{\phi} \mathcal{L}_{\text{fore}}, \qquad \Theta \text{ fixed}. \tag{59}$$

This training strategy enables Graft to incorporate target-domain graph structures into frozen TSFM representations without updating the foundation-model backbone.

B.5.2 Cross-Task Extension to Time-Series Anomaly Detection

To evaluate whether the proposed graph adaptation mechanism generalizes beyond spatial forecasting, we further instantiate Graft for multivariate time-series anomaly detection. Different from spatio-temporal forecasting, where $\mathbf{A}$ describes spatial dependencies among sensors or regions, the anomaly-detection setting constructs $\mathbf{A}^{\text{AD}}$ as an inter-variable dependency graph. Each variable is therefore treated as a graph node, while the graph adaptation architecture remains unchanged.

Given an anomaly-detection input $\mathbf{X}^{\text{AD}}$, the frozen TSFM first extracts temporal tokens, and the same relational graph token alignment, temporal–graph fusion, and CMGA modules produce adapted representations:

$$\widetilde{\mathbf{Z}}^{\text{AD}} = \mathcal{S}_{\phi}\left(\mathbf{Z}^{T,\text{AD}}, \mathbf{X}^{\text{AD}}, \mathbf{A}^{\text{AD}}, \mathbf{M}^{\text{AD}}\right). \tag{60}$$

Instead of using the original forecasting head, we introduce a lightweight trainable reconstruction head $\mathcal{D}_{\psi}^{\text{rec}}$:

$$\hat{\mathbf{X}}^{\text{tar}} = \mathcal{D}_{\psi}^{\text{rec}}\left(\widetilde{\mathbf{Z}}^{\text{AD}}\right), \tag{61}$$

where $\mathbf{X}^{\text{tar}}$ denotes the reconstruction target. The reconstruction objective is defined as

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathbf{X}^{\text{tar}}|} \left\| \mathbf{X}^{\text{tar}} - \hat{\mathbf{X}}^{\text{tar}} \right\|_F^2. \tag{62}$$

During anomaly-detection adaptation, the TSFM backbone remains frozen and only the graph adaptation modules and reconstruction head are optimized:

$$\min_{\phi,\psi} \mathcal{L}_{\text{rec}}, \qquad \Theta \text{ fixed}. \tag{63}$$

The anomaly score is computed from the reconstruction error:

$$s_t = \frac{1}{N} \sum_{i=1}^{N} \left(x_{t,i} - \hat{x}_{t,i}\right)^2. \tag{64}$$

Thus, the anomaly-detection setting changes both the graph semantics and downstream objective while retaining the same frozen TSFM and graph adaptation mechanism. This setting is used to evaluate the cross-task generalization of Graft.

C Datasets, Baselines, and Implementation Details

Datasets. We evaluate Graft in two settings: spatio-temporal forecasting as the primary downstream task and multivariate time-series anomaly detection as a cross-task generalization test.

Spatio-temporal forecasting datasets. We evaluate the proposed Graft on seven widely used spatio-temporal traffic forecasting benchmarks, including PEMS03, PEMS04, PEMS07, PEMS08, PEMS-BAY, METR-LA, and NYC-BIKE. These datasets cover different traffic scenarios, including traffic flow, traffic speed, and bike trajectory forecasting. The PEMS-series datasets Guo et al. (2021) record traffic flow signals from California highway sensor networks at a 5-minute sampling interval, with the

Table 5: Statistics of spatio-temporal traffic benchmark datasets.

Dataset	Category	Sample rate	Regions	Time Points
PEMS03	Traffic Flow	5min	358	26.2k
PEMS04	Traffic Flow	5min	307	17.0k
PEMS07	Traffic Flow	5min	883	28.2k
PEMS08	Traffic Flow	5min	170	17.9k
PEMS-BAY	Traffic Speed	5min	325	52.1k
METR-LA	Traffic Speed	5min	207	34.2k
NYC-BIKE	Bike Trajectory	30min	540	105.2k

number of regions ranging from 170 to 883. PEMS-BAY and METR-LA Li et al. (2018) are traffic speed datasets collected from sensor networks with 325 and 207 regions, respectively. NYC-BIKE Li et al. (2024) contains bike trajectory records with 540 regions and a 30-minute sampling interval. The detailed statistics of these datasets are summarized in Table 5.

Multivariate time-series anomaly detection datasets. To further evaluate the cross-task generalization ability of Graft, we conduct additional experiments on four widely used multivariate time-series anomaly detection benchmarks, including SMD Su et al. (2019), SWaT Goh et al. (2017), WADI Ahmed et al. (2017), and PSM Abdulaal et al. (2021). Table 6 summarizes the detailed statistics of these datasets.

Baselines. For spatio-temporal forecasting, we compare Graft with representative baselines from four categories:

- **Time-Series Expert Models (TSEMs):** Informer Zhou et al. (2021), FEDformer Zhou et al. (2022), DLinear Zeng et al. (2023), PatchTST Nie et al. (2023), and TimeMixer Wang et al. (2024).
- **Spatio-Temporal Expert Models (STEMs):** STID Shao et al. (2022b), STAEformer Liu et al. (2023), and BigST Han et al. (2024).
- **Time-Series Foundation Models (TSFMs):** TimesFM Das et al. (2024), Moirai Woo et al. (2024), and Rose Wang et al. (2025).
- **Spatio-Temporal Foundation Models (STFMs):** UniST Yuan et al. (2024), OpenCity Li et al. (2025), and FactoST-v2 Zhong et al. (2026).

These baselines cover task-specific forecasting architectures, time-series foundation models, and spatio-temporal foundation models.

For multivariate time-series anomaly detection, we compare Graft with eleven advanced multivariate time-series anomaly detection methods. The eleven models are: DAGMM Zong et al. (2018), CAE-M Zhang et al. (2021), OmniAnomaly Su et al. (2019), USAD Audibert et al. (2020), GDN Deng et al. (2021), MSCRED Zhang et al. (2019), TranAD Tuli et al. (2022), MAD-GAN Li et al. (2019), TimesNet Wu et al. (2023), CARLA Darban et al. (2025), CATCH Wu et al. (2025).

Table 6: Statistics of anomaly detection datasets.

Dataset	Domain	Sensors (traces)	Train	Test	Anom. (%)
SMD	Cloud computing	38 (28)	708,405	708,420	4.16%
SWaT	Industrial water treatment	51	47,520	44,991	12.20%
WADI	Industrial water distribution	127	118,795	17,275	5.99%
PSM	Server monitoring	25	132,481	87,841	27.76%

C.1 Implementation Details

The experiments were conducted using PyTorch 2.5.1+cu124 on an 8-core CPU and two NVIDIA RTX A5000 GPUs, each with 24 GB memory. For multi-GPU experiments, GPU memory usage is

Table 7: Common training settings for the main spatio-temporal forecasting experiments.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Batch size	4
Stride	1
Adapter dropout	0.1
Random seed	42 (main experiments)
LR scheduler	StepLR
Step size / decay	10 / 0.5
Graph encoder hidden dim	32

Table 8: Frozen time-series foundation models used in Graft.

Backbone	Pre-training corpus coverage	Parameters
Chronos-2	Real-world time-series data and large-scale synthetic data	120M
Moirai-small	LOTSA: energy, transport, climate, web, sales, nature, economy, and healthcare	13.8M

reported as the aggregate peak memory consumption across both GPUs, using the same measurement protocol for Graft and full TSFM fine-tuning.

Following the standard chronological data partitioning protocol, PEMS03, PEMS04, PEMS07, and PEMS08 are split into training, validation, and test sets with a ratio of 6:2:2, while PEMS-BAY, METR-LA, and NYC-BIKE use a 7:1:2 split. Input and target series are standardized using z-score statistics computed only from the training split, and forecasting metrics are calculated after inverse transformation to the original scale. The few-shot protocol follows Sec. 4.2.

The common training hyperparameters are summarized in Table 7. The maximum number of training epochs varies across datasets and forecasting horizons, and the checkpoint with the lowest validation MAE is used for test evaluation. Exact dataset- and horizon-specific configurations will be released with the source code.

We instantiate Graft with two frozen TSFM backbones, Chronos-2 Ansari et al. (2025) and Moirai-1.1-R-small (hereafter Moirai-small) Woo et al. (2024), as summarized in Table 8. Chronos-2 is a 120M-parameter encoder-only TSFM designed for univariate, multivariate, and covariate-informed forecasting, while Moirai-small is a compact 13.8M-parameter universal forecasting Transformer pre-trained on LOTSA. Across all Graft experiments, the TSFM backbone parameters are kept frozen. For spatio-temporal forecasting, only the lightweight graph adaptation modules are optimized, while anomaly detection additionally optimizes the reconstruction head.

Anomaly detection implementation. For all four anomaly detection datasets, we use a context length of 10, a prediction length of 1, and a stride of 1. The graph adaptation modules and reconstruction head are trained for 10 epochs with a batch size of 16 using AdamW, with a learning rate of 1×10^{-3} and a weight decay of 1×10^{-4}. The adapter dropout is set to 0.1, and the random seed is fixed to 42. For dependency-graph construction, we use the hybrid strategy described in Sec. 4.4, retaining the top-5 dependencies for each variable and considering a maximum temporal lag of 3. The instantaneous and lag-aware dependency components are equally weighted. Adaptive sliding-window standardization is applied before window construction, with normalization windows of 20,000 for SMD, 15,000 for SWaT and WADI, and 30,000 for PSM.

Table 9: Zero-shot forecasting results under short-term and long-term settings. Bold red and underlined blue denote the best and second-best results, respectively. Results marked with * indicate that the corresponding dataset was included in the model pre-training corpus and are excluded from ranking.

Model Type		TSFMs								STFMs					
Dataset	Horizon	Frozen Backbone (Chronos-2)		Rose		TimesFM		Moirai		FactoST-v2		OpenCity		UniST	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PEMS03	12→12	**25.57**	41.73	29.74	47.27	29.71	49.21	28.23*	46.36*	25.70	**40.75**	30.37	47.49	101.87	129.94
	96→96	92.80	135.50	122.93	158.76	108.59	152.74	74.85*	110.78*	**86.90**	**124.62**	125.18	159.23	102.87	137.90
PEMS04	12→12	**32.22**	**49.69**	38.35	57.34	35.00	53.27	34.65*	52.13*	33.13	49.92	39.34*	58.50*	67.91	87.63
	96→96	110.24	155.11	151.39	189.29	127.39	171.13	105.95*	141.40*	**107.15**	**147.60**	153.18*	188.95*	115.76	153.58
PEMS-BAY	12→12	**2.14**	**5.37**	2.72	6.41	6.59	14.99	1.97*	4.69*	2.27	5.46	3.23*	6.91*	14.89	16.92
	96→96	**5.01**	**10.12**	7.02	11.90	14.16	24.01	6.51*	11.70*	6.11	11.56	6.92*	11.79*	9.29	12.72
PEMS07	12→12	**36.97**	**57.46**	44.79	67.33	42.10	64.59	35.64*	50.25*	37.85	57.48	45.18	67.15	104.93	133.65
	96→96	128.73	179.60	172.07	212.94	151.99	205.18	125.91*	169.43*	**126.83**	**173.31**	172.20	211.25	144.67	184.62
PEMS08	12→12	**26.88**	**40.97**	31.70	47.77	29.68	45.18	38.23*	53.12*	27.11	41.18	32.45*	48.42*	73.46	93.14
	96→96	95.66	137.88	129.05	163.15	92.72	**126.10**	119.10*	151.83*	**91.09**	127.28	128.48*	161.75*	104.77	136.21
METR-LA	12→12	5.63	13.17	6.05	14.36	6.59	14.99	5.55	13.79	**5.32**	**12.82**	4.30*	8.37*	24.33	29.31
	96→96	12.47	22.41	12.96	23.61	14.16	24.01	12.87	22.19	**10.87**	**21.49**	10.85*	17.60*	25.88	30.19
NYC-BIKE	12→12	1.54	7.72	2.31	8.96	**1.11**	**7.12**	2.46	8.79	2.07	8.32	3.30	9.37	5.95	11.34
	96→96	**1.01**	**5.52**	2.53	9.21	1.26	6.77	2.87	9.36	1.67	7.49	5.41	11.32	7.35	13.85

D Complete Quantitative Results

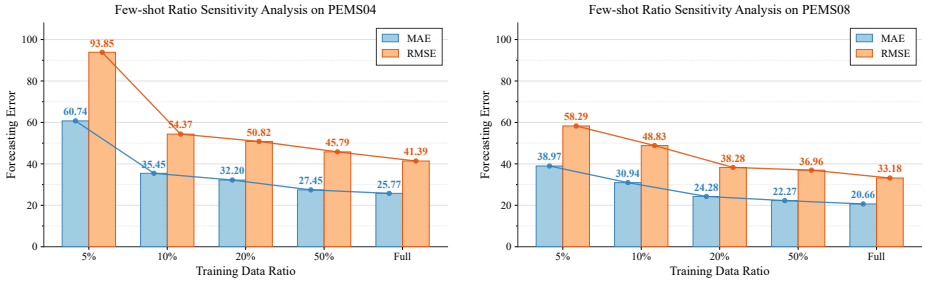
D.1 Full Zero-Shot Comparison

Analysis. These zero-shot results show that dedicated spatio-temporal pre-training does not yield a uniform advantage across heterogeneous target graphs. The frozen Chronos-2 backbone is competitive with, and on several datasets better than, existing STFMs despite not using target-domain graph structure during pre-training. The contrast is particularly informative because temporal regularities such as trend, seasonality, and local continuity can transfer across datasets, whereas graph topology and spatial interactions are strongly domain dependent. This observation motivates the design choice of Graft: rather than binding temporal and spatial knowledge inside a jointly pre-trained backbone, it preserves the transferable temporal prior and introduces graph structure only when adapting to the target domain. Results marked with * are excluded from ranking because the corresponding dataset appears in the model's pre-training corpus.

E Additional Analyses

E.1 Few-Shot Ratio Sensitivity

We vary the proportion of labeled target-domain training windows to evaluate how quickly Graft benefits from additional supervision and whether its gains saturate as more data become available. Figure 6 shows that Graft benefits most strongly from the first increase in supervision. On PEMS04,



(a) PEMS04 (b) PEMS08

Figure 6: Few-shot ratio sensitivity analysis on PEMS04 and PEMS08 under the 96-to-96 forecasting setting.

increasing the training ratio from 5% to 10% reduces MAE from 60.74 to 35.45 and RMSE from 93.85 to 54.37; on PEMS08, MAE/RMSE decrease from 38.97/58.29 to 30.94/48.83. Performance continues to improve at 20%, 50%, and full-data training, but the marginal gains become progressively smaller. This saturation trend indicates that the graph adaptation modules can effectively exploit target-domain spatial structure from a relatively small amount of labeled data, which is consistent with the few-shot objective of Graft.

E.2 Zero-to-Few-Shot Adaptation Gains

We further compare raw frozen-backbone inference with 10% few-shot Graft adaptation to quantify how much performance is recovered by learning only the target-domain graph adaptation modules.

Table 10: Relative improvement from the zero-shot frozen backbone to few-shot Graft, where the few-shot setting uses only 10% labeled training data to optimize the lightweight graph adaptation modules while keeping the backbone frozen.

Dataset	Horizon	Zero-shot Backbone		Few-shot Graft		Improvement	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
PEMS03	12→12	25.57	41.73	16.66	26.47	34.85%	36.57%
	96→96	92.80	135.50	26.85	43.55	71.07%	67.86%
PEMS04	12→12	32.22	49.69	22.49	35.24	30.20%	29.08%
	96→96	110.24	155.11	35.45	54.37	67.84%	64.95%
PEMS-BAY	12→12	2.14	5.37	1.82	4.34	14.95%	19.18%
	96→96	5.01	10.12	2.86	6.21	42.91%	38.64%
PEMS07	12→12	36.97	57.46	23.49	37.57	36.46%	34.62%
	96→96	128.73	179.60	34.78	58.69	72.98%	67.32%
PEMS08	12→12	26.88	40.97	17.81	28.23	33.74%	31.10%
	96→96	95.66	137.88	30.94	48.83	67.66%	64.59%
METR-LA	12→12	5.63	13.17	5.12	12.66	9.06%	3.87%
	96→96	12.47	22.41	10.69	20.10	14.27%	10.31%
NYC-BIKE	12→12	1.54	7.72	0.96	5.28	37.66%	31.61%
	96→96	1.01	5.52	0.82	4.32	18.81%	21.74%

The relative-improvement results further quantify the value of adaptation over raw frozen-backbone inference. Long-horizon improvements are especially large on the PEMS datasets: MAE reductions reach 71.07% on PEMS03, 67.84% on PEMS04, 72.98% on PEMS07, and 67.66% on PEMS08. The gains are smaller on METR-LA, where the zero-shot backbone is already relatively competitive, but remain positive for both MAE and RMSE. This pattern suggests that Graft is most beneficial when the frozen temporal prior alone cannot capture the target graph structure, while avoiding unnecessary backbone updates when the temporal prior is already strong.

E.3 Node-Wise Error Visualization

Average forecasting metrics can hide heterogeneous behavior across sensors. We therefore inspect node–horizon error maps to determine whether the gains of Graft are broadly distributed or concentrated on a small subset of nodes. Figure 7 provides a node-level view of where the forecasting gains arise. For the Chronos-2 zero-shot backbone, errors generally increase as the prediction horizon grows, and several sensors form persistent high-error regions, indicating that temporal priors alone are insufficient to account for node-specific spatial interactions. After graph adaptation, the Graft heatmaps become consistently darker over most sensors and horizons. The MAE-reduction maps are predominantly positive, with particularly clear gains at longer horizons. This supports the quantitative results by showing that Graft improves not only the average error but also the spatial distribution of forecasting accuracy across nodes.

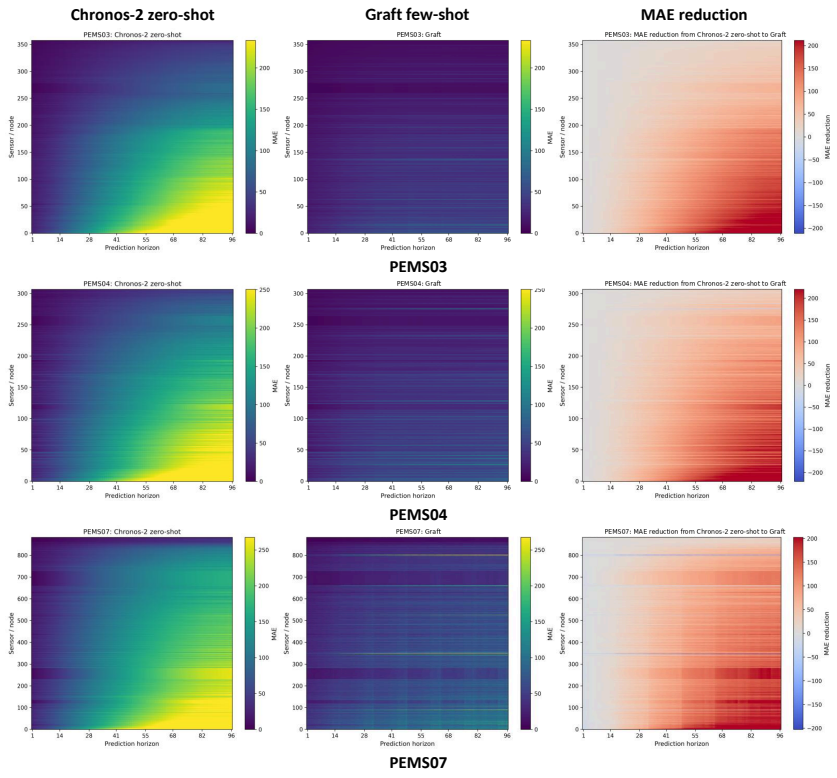


Figure 7: Node-wise spatial error visualization under the 10% few-shot setting. Each row corresponds to one dataset, and the three columns show the node-horizon MAE heatmaps of Chronos-2 zero-shot forecasting, Graft few-shot forecasting, and the MAE reduction from Chronos-2 zero-shot to Graft, respectively. The horizontal axis denotes the prediction horizon, and the vertical axis denotes traffic sensors sorted by the average zero-shot error.

Table 11: Effectiveness of graph adaptation for cross-task anomaly detection on WADI and PSM.

Method	**WADI**			**PSM**		
	$F1_T$	F1	$F1_{PA}$	$F1_T$	F1	$F1_{PA}$
Frozen TSFM + Recon. Head	0.480	0.423	0.727	0.481	0.468	0.916
Graft	**0.521**	**0.466**	**0.766**	**0.571**	**0.516**	**0.952**

E.4 Effect of graph adaptation

To further examine whether the performance gains on anomaly detection originate from graph adaptation rather than the frozen temporal backbone alone, we compare Graft with a temporal-only baseline that directly attaches a reconstruction head to the frozen TSFM. As shown in Table 11, Graft consistently improves all evaluation metrics on both WADI and PSM. Compared with the temporal-only baseline, Graft improves $F1_T$, F1, and $F1_{PA}$ by 0.041, 0.043, and 0.039 on WADI, and by 0.090, 0.048, and 0.036 on PSM, respectively. These consistent gains demonstrate that modeling inter-variable dependencies through the constructed relational graph substantially improves the detection of temporally correlated abnormal behaviors, further supporting the effectiveness of Graft when transferred to a different downstream task and graph semantics.

E.5 Robustness Across Random Seeds

To evaluate the robustness of Graft to training randomness, we repeat the 10% few-shot experiments with Chronos-2 under the $96 \rightarrow 96$ forecasting setting using three random seeds (41, 42, and 43). We

select PEMS03, PEMS04, and PEMS08 as representative benchmarks and keep all other training configurations, data splits, few-shot subsets, and graph structures unchanged across runs. Table 12 reports the individual results for each seed together with the mean and standard deviation over the three runs.

Table 12: Robustness of Graft across three random seeds under the 10% few-shot and $96 \rightarrow 96$ forecasting setting with Chronos-2.

Dataset	Seed 41		Seed 42		Seed 43		Mean $\pm$ Std	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PEMS03	26.43	43.04	26.85	43.55	26.67	43.60	26.65 ± 0.21	43.40 ± 0.31
PEMS04	34.34	53.41	35.45	54.37	35.22	55.10	35.00 ± 0.58	54.29 ± 0.85
PEMS08	30.73	48.53	30.94	48.83	30.79	48.84	30.82 ± 0.11	48.73 ± 0.18

The results exhibit consistently small variations across different random seeds. The standard deviation of MAE remains below 0.6 on all three datasets, while the corresponding RMSE deviation remains below 0.9. In addition, the seed-42 results reported in the main experiments are close to the three-run averages and are not consistently the best-performing runs. These results indicate that the performance of Graft is robust to training randomness and is not dependent on a particular random initialization.