CAUSALITY-INSPIRED SPATIO-TEMPORAL ATTENTION FOR FORECASTING ON EVOLVING GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Urban forecasting is particularly critical when infrastructure changes, as road-network and communication entities may be added, removed, or disrupted, altering graph topology and data distributions. Most spatio-temporal forecasters assume fixed node sets and connectivity. Newly activated entities lack sufficient long-term observations, while frequent retraining may be impractical for time-sensitive traffic control and emergency recovery. We therefore study multi-step forecasting on evolving graphs and propose DyCSTA, a spatio-temporal forecasting framework for multi-step prediction on evolving graphs. DyCSTA retains usable history for persistent nodes, models newly arriving nodes from their current observations, reconstructs dependencies on the current graph, and uses latent context to assist prediction. Its predictive representation split separates forecast-relevant dynamics from context-related variation, while a model-defined replacement branch regularizes context conditioning during training. Experiments on traffic, communication, and other evolving systems evaluate changing node sets, new-node forecasting, and all prediction horizons. DyCSTA achieves its strongest results on the two flow datasets and remains competitive in the other domains.

1 INTRODUCTION

Spatio-temporal forecasting supports transportation, communication, environmental, and energy applications. Existing graph models capture temporal and spatial relations Li et al. (2018); Yu et al. (2018); Wu et al. (2019); Bai et al. (2020), but commonly assume fixed nodes and connectivity during training and inference.

This assumption fails when road or base-station infrastructure changes. Persistent nodes retain dynamics from the preceding stage, whereas arriving nodes begin with limited current-stage observations. A forecaster must handle both histories without retraining after every topology change.

Reusing all history can carry information invalidated by changed connections, while discarding it wastes persistent-node dynamics. The model must retain stable history, incorporate arriving nodes, and rebuild relations on the current graph.

Dynamic-graph methods update representations or parameters, continual forecasters preserve history through replay or memory Chen et al. (2021); Chen & Liang (2025); Zhang et al. (2025a); Liu & Zhang (2026), and inductive methods address entities with little history Zheng et al. (2025); Zhang et al. (2026). We combine these requirements when both node sets and connectivity change between stages.

Causality-inspired forecasting and dynamic-graph methods separate prediction-relevant factors from context variation, learn latent codes, or recombine representations during training Sui et al. (2022); Zhang et al. (2022); Xia et al. (2023); Zhang et al. (2024). We use a context representation to infer a latent code while assigning the primary forecasting role to the forecast-relevant component.

We design DyCSTA to combine selective state transfer, current-graph spatio-temporal encoding, and latent-context-assisted prediction. Our contributions are:

- We formulate stage-wise multi-step forecasting with node arrival, departure, and connectivity changes. A stability-aware module transfers selected persistent-node states, initializes

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.

arriving nodes from the current window, and adapts propagation before current-graph encoding.

- We introduce a predictive decomposition in which soft latent-context prototypes condition a forecast-relevant representation. Detached prototype correction and a model-defined context-replacement objective regularize this conditioning.
- We evaluate four evolving-graph datasets, including arriving-node and horizon-wise analyses. The strongest evidence is on the two flow datasets, with competitive performance in the other domains.

2 Related Work

Spatio-temporal forecasting. DCRNN, STGCN, Graph WaveNet, and AGCRN combine temporal modeling with predefined or adaptive propagation Li et al. (2018); Yu et al. (2018); Wu et al. (2019); Bai et al. (2020); attention-based models capture long-range or time-varying relations Zheng et al. (2020); Li et al. (2023); Jiang et al. (2023); Liu et al. (2023). Our setting additionally includes node arrival, departure, and changed connectivity.

Dynamic graphs and continual forecasting. EvolveGCN evolves graph-convolution parameters and DySAT learns structural and temporal attention Pareja et al. (2020); Sankar et al. (2020); later methods model influential nodes, temporal subtrees, or scalable transformations Tan et al. (2025); Nguyen & Ta (2025); Wu et al. (2026). Continual forecasters retain patterns through replay or memory Chen et al. (2021); Chen & Liang (2025); Zhang et al. (2025a); Liu & Zhang (2026). DyCSTA combines retained history with current-graph context when both nodes and connectivity change.

Causality-inspired representation learning. SCMs characterize functional relations beyond observational association Pearl (2000). CAL separates causal and shortcut representations, DIDA recombines invariant and variant patterns, and CaST and Caformer model environment-related factors Sui et al. (2022); Zhang et al. (2022); Xia et al. (2023); Zhang et al. (2024). DyCSTA adapts this representational separation to predictive latent context on evolving graphs.

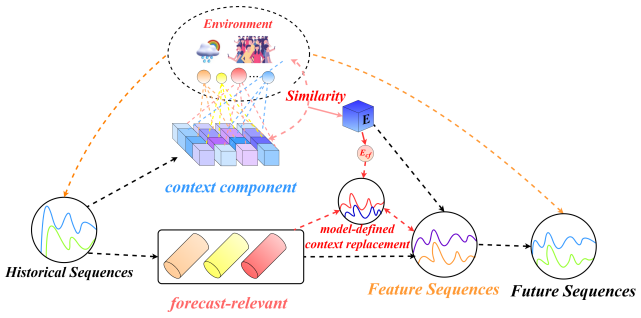

Figure 1: Predictive relations among historical representations, latent context, and future states.

3 Problem Formulation

Definition 1 (Evolving Graph). Let $\mathcal{G} = \{G_1, \ldots, G_T\}$ be a temporally ordered graph sequence, where

$$G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{A}_t). \tag{1}$$

$\mathcal{V}_t$, $\mathcal{E}_t$, and $\mathbf{A}_t \in \mathbb{R}^{N_t \times N_t}$ are the active node set, edge set, and adjacency matrix. Node and edge changes allow both N_t and propagation structure to vary across stages.

Definition 2 (Spatio-Temporal Forecasting). Given the most recent P observations on G_t, we estimate the next Q states by

$$\widehat{\mathbf{Y}}_{\tau+1:\tau+Q} = f_\theta(\mathbf{X}^{(t)}_{\tau-P+1:\tau}, G_t). \tag{2}$$

Only nodes active in G_t participate in propagation, loss, and evaluation; their current topology is observed at prediction time.

Causality-inspired predictive perspective. We use causality-inspired decomposition as an inductive bias for separating forecast-relevant dynamics from contextual variation. Historical observations X produce a forecast-relevant component C and a context representation T; T infers a latent context code E that conditions C for predicting Y. Thus, C, T, and E are learned predictive roles and codes, and replacing E while retaining C defines a model-defined context replacement.

4 METHOD

4.1 MODEL OVERVIEW

DyCSTA transfers states for selected persistent nodes, incorporates arriving nodes from the current window, and reconstructs dependencies on the current graph. Its encoder separates forecast-relevant and context representations, with the latter inferring a latent code for the former. Figure 2 presents the architecture.

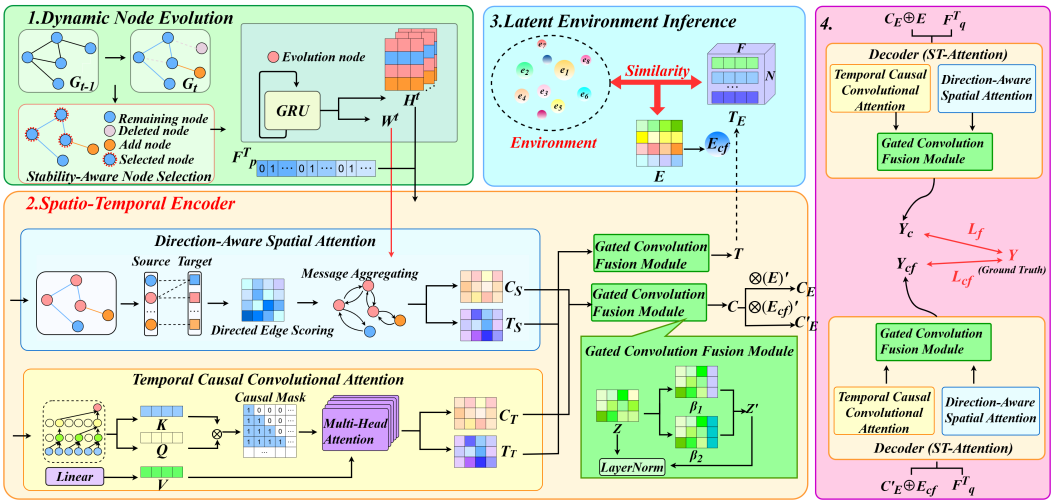

Figure 2: The DyCSTA architecture, comprising cross-topology dynamic evolution, current-graph spatio-temporal encoding, latent context inference, and multi-step forecasting.

4.2 DYNAMIC NODE EVOLUTION

Unequal historical availability makes wholesale inheritance and complete removal undesirable. DyCSTA retains dynamics from stable persistent nodes and admits arriving nodes through a common current-stage window. Let $\mathcal{V}_t^{\text{per}}$, $\mathcal{V}_t^{\text{new}}$, and $\mathcal{V}_t^{\text{del}}$ denote the persistent, arriving, and departing node sets. For persistent node v, the cross-stage change is

$$d_v = W_1\left(p_v^{t-1}, p_v^t\right). \tag{3}$$

Here, p_v^{t-1} and p_v^t are empirical distributions, and d_v measures change. Nodes with smaller d_v enter evolution pool $\mathcal{S}_t$ first; high-activity arriving nodes fill any shortage.

Let $\mathbf{X}_{\mathcal{S}}^t$ be the projected P-step pool input. A shared GRU produces evolved states $\mathbf{H}^t$ and a candidate propagation update $\Delta\mathbf{W}^t$. The current propagation parameter is

$$\mathbf{W}^t = \mathbf{W}_0 + \eta_t \Delta\mathbf{W}^t. \tag{4}$$

Here, $\mathbf{W}_0$ is the learnable base parameter and η_t is an adaptive gate from pooled current-stage observations and the candidate update. Only corresponding persistent nodes receive $\mathbf{H}^t$; arriving nodes use the current P-step window. All active nodes then enter current-graph encoding, with a gated residual injecting $\mathbf{H}^t$ into the forecast-relevant representation.

This construction separates two sources of information that are otherwise mixed by a single recurrent state. The inherited state carries slowly changing node-specific dynamics across stages, while the

current window supplies information available to every active node, including a newly arriving node. The propagation update is shared at the stage level rather than assigned to a fixed node index, so it can be applied after the active set changes. State transfer and graph reconstruction are therefore complementary: the former preserves usable history for persistent nodes, and the latter prevents stale connectivity from being reused for the current graph.

4.3 SPATIO-TEMPORAL ENCODER

Dynamic evolution links adjacent snapshots, while predictive relations are rebuilt on the changed graph. Let $\mathbf{X}$ denote projected active-node inputs and $\mathbf{F}_P^T$ the weekday and time-of-day embeddings; the superscript T denotes temporal embedding, not transposition.

Temporal causal convolutional attention. The temporal branch concatenates $\mathbf{X}$ and $\mathbf{F}_P^T$, uses left-padded causal convolutions for query $\mathbf{Q}$ and key $\mathbf{K}$, and projects value $\mathbf{V}$:

$$\mathbf{A}_T = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}\right). \tag{5}$$

Here, $\mathbf{A}_T$ is temporal attention, d is the head dimension, and the mask M permits only current or preceding positions. Two normalized role weights split the output into $\mathbf{C}_T$ and $\mathbf{T}_T$.

Direction-aware spatial attention. To model asymmetric propagation, the spatial branch transforms the source and target of current edge (i, j) separately and computes

$$a_{ij} = \sigma\big(g_e\big(\mathbf{h}_i^{\mathrm{src}} \,\|\, \mathbf{h}_j^{\mathrm{tgt}}\big)\big). \tag{6}$$

Here, $\mathbf{h}_i^{\mathrm{src}}$ and $\mathbf{h}_j^{\mathrm{tgt}}$ are transformed endpoint states, g_e scores their concatenation, and σ yields directed weight a_{ij}. The current edges, a_{ij}, and $\mathbf{W}^t$ produce $\mathbf{C}_S$ and $\mathbf{T}_S$.

Gated convolutional fusion. Two independent modules fuse components with the same role:

$$\mathbf{C} = \mathrm{GCF}_C(\mathbf{C}_T, \mathbf{C}_S). \tag{7}$$

$$\mathbf{T} = \mathrm{GCF}_T(\mathbf{T}_T, \mathbf{T}_S). \tag{8}$$

Here, $\mathbf{C}$ and $\mathbf{T}$ are the fused forecast-relevant and context representations. In Figure 2, $\mathbf{Z}$ enters content and gating branches $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, yielding residual normalized output $\mathbf{Z}'$; $\mathbf{H}^t$ is injected into $\mathbf{C}$.

4.4 LATENT CONTEXT INFERENCE

The design assigns $\mathbf{C}$ the primary forecasting role and maps $\mathbf{T}$ to a soft mixture of learned context prototypes that conditions $\mathbf{C}$.

Pooling $\mathbf{T}$ and $\mathbf{C}$ gives $\bar{\mathbf{T}}_i$ and $\bar{\mathbf{C}}_i$. Let $\{\mathbf{e}_k\}_{k=1}^{K_E}$ be learnable context prototypes. A stop-gradient projection of $\bar{\mathbf{C}}_i$ corrects the similarity to prototype k:

$$s_{ik} = \cos\big(g_T(\bar{\mathbf{T}}_i), \mathbf{e}_k\big) - \lambda_o \cos\big(g_C(\mathrm{sg}(\bar{\mathbf{C}}_i)), \mathbf{e}_k\big). \tag{9}$$

The stop-gradient confines this correction term to the prototype assignment; the main and auxiliary paths still backpropagate through $\mathbf{C}$. Here, g_T and g_C are projection maps and λ_o controls correction strength. Define $\mathbf{T}_{E,i} = [T_{E,i1}, \ldots, T_{E,iK_E}]$:

$$T_{E,ik} = \frac{\exp(s_{ik}/\tau_e)}{\sum_{j=1}^{K_E} \exp(s_{ij}/\tau_e)}. \tag{10}$$

Here, $T_{E,ik}$ is the prototype weight and τ_e is the temperature. The implementation averages assignments over temporal and node dimensions before forming the prototype mixture and broadcasts the resulting code. The latent context code is

$$\mathbf{E}_i = g_E\left(\sum_{k=1}^{K_E} T_{E,ik}\mathbf{e}_k\right). \tag{11}$$

Here, g_E maps the prototype mixture to context code $\mathbf{E}_i$. It modulates $\mathbf{C}$ as

$$\mathbf{C}_E = \mathbf{C} \odot [1 + \tanh(m(\mathbf{E}))] . \tag{12}$$

Here, m is the modulation network and $\odot$ is element-wise multiplication. The decoder broadcasts the projected code, combines $\mathbf{C}_E \oplus \mathbf{E}$ with future embedding $\mathbf{F}_Q^T$, and outputs $\mathbf{Y}_c = \widehat{\mathbf{Y}}$. It does not perform a second predictive decomposition.

The conditioning is applied after the two encoder roles have been fused, so the context code changes the amplitude of the forecast-relevant features without replacing their temporal and spatial content. This placement also makes the factual and replacement paths comparable: they share $\mathbf{C}$ and decoder parameters, and differ in the code supplied to the modulation gate. The resulting comparison measures the contribution of context conditioning under the same representation and output space.

4.5 Loss Function

During training, an auxiliary replacement context code $\mathbf{E}_{\mathrm{cf}}$ is selected from a different prototype with the closest running error estimate. It modulates $\mathbf{C}$ into $\mathbf{C}'_E$ and uses the same decoder to produce $\mathbf{Y}_{\mathrm{cf}}$. Both paths use masked MAE:

$$\mathcal{L}_f = \frac{1}{M_y} \sum_{m=1}^{M_y} |Y_{c,m} - Y_m| . \tag{13}$$

$$\mathcal{L}_{\mathrm{cf}}^{\mathrm{err}} = \frac{1}{M_y} \sum_{m=1}^{M_y} |Y_{\mathrm{cf},m} - Y_m| . \tag{14}$$

Here, $\mathbf{Y}$ is ground truth, M_y counts nonzero targets, and $\mathcal{L}_f$ and $\mathcal{L}_{\mathrm{cf}}^{\mathrm{err}}$ are factual and replacement errors. Their margin loss is

$$\mathcal{L}_{\mathrm{cf}} = [\mu - (\mathcal{L}_{\mathrm{cf}}^{\mathrm{err}} - \mathcal{L}_f)]_+ . \tag{15}$$

Here, μ is the decreasing margin and $[x]_+ = \max(x, 0)$. When violated, the hinge term encourages replacement error to exceed factual error by at least μ. The prototype-usage loss encourages balanced batch-level usage:

$$\mathcal{L}_{\mathrm{use}} = \frac{1}{K_E} \left\| \frac{1}{B} \sum_{i=1}^{B} \mathbf{T}_{E,i} - \frac{1}{K_E} \mathbf{1}_{K_E} \right\|_2^2 . \tag{16}$$

Here, B is batch size and $\mathbf{1}_{K_E}$ is all ones. Applied to the batch-mean response, the loss promotes aggregate prototype coverage rather than sample-level diversity. The total objective is

$$\mathcal{L} = \mathcal{L}_f + \alpha \mathcal{L}_{\mathrm{cf}} + \beta \mathcal{L}_{\mathrm{use}}. \tag{17}$$

Here, α and β weight the auxiliary losses. The auxiliary branch uses the selected replacement code and sigmoid gate, whereas the factual branch uses the tanh gate in Eq. (12); only the factual path is used at inference.

5 Experiments

We evaluate accuracy, component contributions, new-node behavior, representation separation, and horizon-wise performance; protocols and extended results appear in the appendix.

5.1 Experimental Setup

Datasets. We evaluate PEMS-Stream road traffic, JiNan wireless traffic, Air-Stream air quality, and Energy-Stream wind power. All tasks use 12 historical steps to predict the next 12; metrics cover active nodes and mask zero targets. Dataset dates, snapshots, graph statistics, preprocessing, and splits appear in the appendix. "PEMS," "Air," and "Energy" abbreviate the corresponding Stream datasets.

Baselines. We compare DMLSA and DynaSTy Zhang et al. (2025b); Banerji & Berger-Wolf (2026); DyAtGNN, DySubTree, and ScaDyG Tan et al. (2025); Nguyen & Ta (2025); Wu et al. (2026); and

Table 1: Multi-step forecasting results on four evolving-graph datasets. Lower is better; bold and underlined values denote the best and second-best results within each dataset–metric block. The average is computed over all 12 forecast horizons, while only horizons 3, 6, and 12 are displayed. MAPE is reported in percent.

Dataset		***PEMS-Stream*** $655 \rightarrow \cdots \rightarrow 871$				***JiNan*** $8 \rightarrow \cdots \rightarrow 11$				***Air-Stream*** $1087 \rightarrow \cdots \rightarrow 1202$				***Energy-Stream*** $103 \rightarrow \cdots \rightarrow 134$			
Method	**Metric**	3	6	12	**Avg.**	3	6	12	**Avg.**	3	6	12	**Avg.**	3	6	12	**Avg.**
DMLSA (2025)	MAE	21.05	22.29	27.42	21.97	215.31	260.63	308.14	254.21	13.79	17.43	20.56	16.45	1.74	2.22	2.15	2.84
	RMSE	54.64	53.69	60.88	52.56	327.73	387.42	453.05	385.97	24.00	30.06	33.98	28.63	4.29	4.99	3.46	7.16
	MAPE	51.27	49.35	63.62	48.71	27.80	34.92	41.53	33.04	25.49	30.67	37.94	29.65	31.06	37.68	29.16	46.61
DySubTree (2025)	MAE	13.95	16.58	22.37	17.08	206.43	273.58	321.99	256.64	12.87	17.12	20.43	16.20	2.89	3.10	4.27	3.62
	RMSE	23.97	28.45	37.22	29.45	310.52	389.90	447.44	376.75	22.69	28.99	33.32	28.02	4.12	4.21	5.38	4.84
	MAPE	16.83	19.13	25.16	19.75	29.42	41.46	48.75	37.58	22.02	29.84	37.67	28.49	53.14	53.24	67.64	61.28
DyAtGNN (2025)	MAE	13.06	14.81	18.12	15.00	216.04	253.00	294.98	250.10	12.83	16.75	19.82	15.86	3.00	3.39	4.30	3.49
	RMSE	22.54	25.67	31.25	25.91	327.15	385.54	445.66	378.94	22.45	28.55	33.68	27.20	4.23	4.62	5.58	4.73
	MAPE	21.24	22.39	23.53	22.63	29.02	33.21	38.27	32.94	21.72	**28.48**	**33.71**	**26.96**	56.02	61.28	73.65	62.61
EAC (2025)	MAE	17.47	19.58	24.75	20.07	202.59	246.34	312.90	244.79	14.92	19.28	22.12	18.26	1.41	1.65	2.64	1.65
	RMSE	30.62	33.65	40.70	34.21	310.25	379.63	450.81	374.46	24.06	30.10	34.49	29.24	1.88	2.11	3.22	2.20
	MAPE	25.27	28.19	36.04	29.23	26.67	31.62	44.30	32.56	31.37	41.38	47.11	38.88	52.71	49.42	69.12	48.43
STRAP (2025)	MAE	14.69	17.06	22.98	17.66	195.85	242.18	311.11	244.26	13.58	18.20	21.56	17.16	1.68	2.09	2.96	1.96
	RMSE	24.74	28.91	38.06	30.02	304.32	372.49	447.99	372.47	22.95	29.38	33.79	28.38	2.19	2.66	3.58	2.56
	MAPE	20.38	21.52	28.49	22.90	25.82	31.93	43.67	32.96	25.86	35.68	43.63	33.86	61.24	69.06	84.24	61.92
COIN-GNN (2025)	MAE	13.98	16.24	21.26	16.62	197.32	245.53	287.74	242.49	16.72	20.66	22.71	19.69	1.24	1.38	2.15	1.47
	RMSE	23.74	27.63	35.27	28.38	303.30	374.07	441.89	371.73	25.34	31.12	35.23	30.28	1.72	1.82	2.72	2.00
	MAPE	17.42	19.71	25.62	20.22	27.31	34.34	38.19	33.68	37.82	45.68	47.81	43.44	20.93	19.06	27.06	20.37
DynaSTy (2026)	MAE	13.71	15.57	19.76	15.87	197.32	245.80	301.26	242.02	13.00	17.27	20.24	16.21	0.73	1.55	2.07	1.46
	RMSE	23.64	26.92	33.51	27.50	310.79	379.47	445.18	375.34	22.92	29.23	33.43	28.17	1.11	2.06	2.75	2.07
	MAPE	16.46	18.30	22.78	18.67	25.52	33.31	42.99	32.87	22.15	30.02	36.83	28.56	7.30	17.75	20.83	16.35
ScaDyG (2026)	MAE	16.22	19.55	27.24	20.38	**193.34**	247.42	318.66	245.51	13.59	18.29	21.71	17.19	**0.65**	**1.12**	2.07	**1.17**
	RMSE	**19.55**	32.90	44.59	34.61	313.46	398.54	497.13	399.58	24.09	30.62	35.18	29.58	**1.02**	**1.57**	2.62	**1.74**
	MAPE	19.18	22.58	31.09	23.65	**22.89**	30.16	40.57	30.00	23.61	32.67	40.11	30.94	**6.95**	**12.32**	23.47	**12.89**
STBP (2026)	MAE	14.29	15.98	19.62	16.23	201.48	239.97	297.20	240.12	13.83	19.56	22.80	18.13	1.16	1.85	2.81	1.91
	RMSE	27.71	31.28	36.12	31.43	**301.34**	364.03	436.13	363.67	23.71	32.51	37.94	31.33	1.72	2.62	3.82	2.79
	MAPE	21.83	23.41	27.47	23.84	28.48	32.64	41.07	33.02	24.48	35.23	41.99	32.90	19.33	29.63	44.44	30.50
DyCSTA	MAE	**12.57**	**13.82**	**16.29**	**13.97**	195.19	**222.75**	**257.01**	**221.31**	**12.39**	**16.70**	**19.73**	**15.68**	0.92	1.18	**1.79**	1.26
	RMSE	21.97	**24.28**	**28.36**	**24.41**	309.67	**349.19**	**396.56**	**347.07**	**22.21**	**28.55**	**32.98**	**26.91**	1.31	1.67	**2.41**	1.76
	MAPE	**15.09**	**16.28**	**19.11**	**16.49**	26.06	**29.45**	**34.40**	**29.34**	**21.04**	29.03	35.32	27.45	10.89	13.23	**17.41**	13.51

EAC, STRAP, STBP, and COIN-GNN Chen & Liang (2025); Zhang et al. (2025a); Liu & Zhang (2026); Zheng et al. (2025). All use the same snapshots, horizons, splits, metrics, and validation-based checkpoint selection.

Implementation. We use an NVIDIA RTX 3090 GPU, train for at most 100 epochs with batch size 32, and stop early after 15 epochs without validation improvement.The relevant code is included in the supplementary material to support reproducibility.

5.2 Main Results

Table 1 reports horizons 3, 6, 12, and their average; complete comparisons appear in the appendix.

DyCSTA obtains the best 12-step average on all three metrics for PEMS-Stream and JiNan, with short-range RMSE and MAPE exceptions at individual horizons. On Air-Stream it leads the average absolute-error metrics while remaining competitive in percentage error; on Energy-Stream it ranks second to ScaDyG. The advantage is most consistent on the flow datasets, where node arrivals and departures directly alter the test-stage graph, while the environmental datasets show more variable gains. The complete horizon curves show that these comparisons are not determined by one endpoint.

The comparison also separates the evolving-graph mechanism from the forecasting horizon. A short horizon can be dominated by recent observations, whereas longer horizons place more weight on retained state, current-graph dependencies, and context-conditioned decoding. The average therefore summarizes all 12 targets, while the horizon-wise analyses expose where each method gains or loses relative to the evolving dynamics.

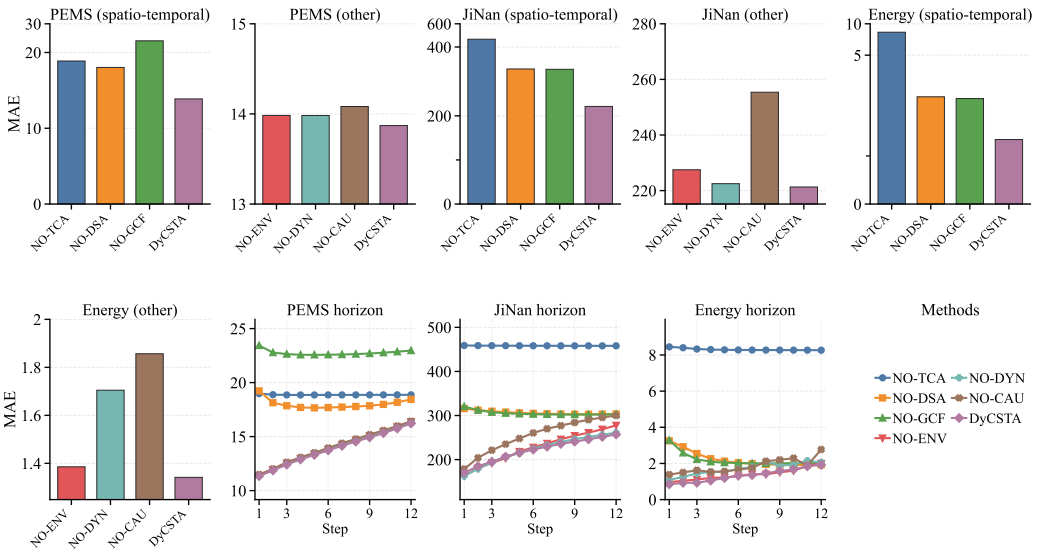

Figure 3: MAE ablations for seven variants on PEMS-Stream, JiNan, and Energy-Stream. Bars show the 12-step average and line panels show horizon-wise MAE. The spatio-temporal variants remove TCA, DSA, or GCF, while the auxiliary variants remove environment assistance, dynamic evolution, or the causality-inspired representation branch. Lower is better.

The dataset ordering is also informative. PEMS-Stream and JiNan are flow datasets in which the target evolves with the active graph, so retaining persistent-node history and rebuilding current dependencies address the same source of distribution change. Air-Stream and Energy-Stream contain stronger measurement-specific variation; the competitive results on these datasets indicate transfer beyond traffic, while the smaller margins motivate the per-dataset analysis in the appendix.

5.3 Ablation Study

We evaluate seven variants on PEMS-Stream, JiNan, and Energy-Stream, representing large, small, and medium evolving graphs, respectively. NO-TCA, NO-DSA, NO-GCF, NO-ENV, NO-DYN, and NO-CAU remove temporal attention, direction-aware spatial attention, gated fusion, context assistance, dynamic evolution, and the causality-inspired representation branch, respectively. All variants use the same splits, horizons, and training protocol.

Figure 3 reports MAE: DyCSTA achieves the lowest 12-step average on all three datasets, while removing any tested component increases error, with the largest gap varying by dataset. The dynamic-evolution gap is smaller on JiNan, whose snapshots contain only 8–11 active base stations, and larger on Energy-Stream. RMSE and MAPE results are provided in the appendix.

5.4 Dynamic-Node Arrival Generalization

We evaluate all, existing, and newly arriving PEMS-Stream nodes over the same splits and 12 horizons. Figure 4 shows DyCSTA and five baselines selected by overall MAE; the appendix contains all methods and metrics.DyCSTA achieves the lowest errors on all and existing nodes and remains competitive on arriving nodes as the horizon grows. The overall result is therefore not explained solely by persistent-node predictions.

5.5 Predictive Representation Separation and Environment Assistance

We assess the learned split on PEMS-Stream using forecast-relevant predictor C_0, context-assisted branch C_E, and context branch T_E. C_E and T_E use the same frozen encoder and identically initialized probe decoders; the appendix gives the protocol.Figure 5 shows T_E/C_E error ratios above one under matched probes. C_E remains below the C_0 reference, with larger later-horizon gains. These

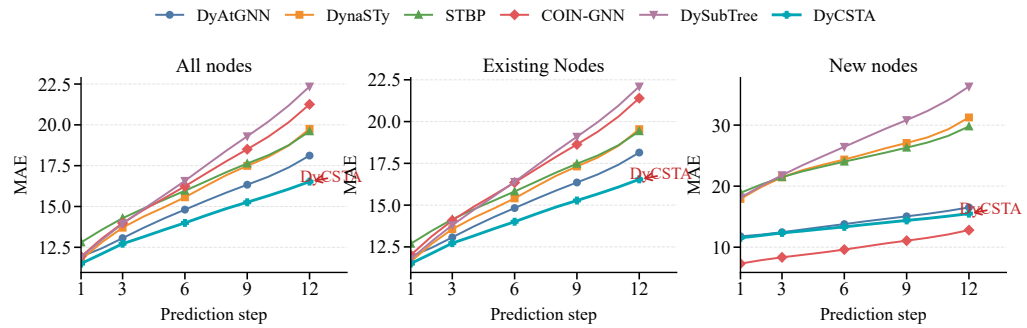

Figure 4: Horizon-wise MAE on all, existing, and newly arriving PEMS-Stream nodes for DyCSTA and five strong baselines selected by overall MAE. Red arrows mark DyCSTA. The appendix reports all evaluated methods and the complete MAE, RMSE, and MAPE comparisons.

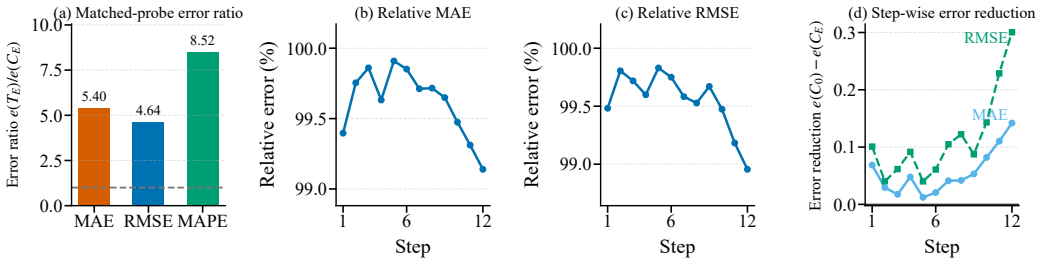

Figure 5: Matched-probe analysis of predictive representation separation and environment assistance. The ratio is $e(T_E)/e(C_E)$ using 12-step average errors, so values above one mean that C_E has lower error. The remaining panels report C_E relative to C_0 and the horizon-wise error reduction.

comparisons support the predictive roles of the two representations and latent-context conditioning.

5.6 Forecast-Range Performance

We compare all 12 steps with one fixed strong baseline per dataset–metric pair, selected by its 12-step average. Figure 6 summarizes aggregate trends and long-range behavior; dataset-wise curves and comparator identities appear in the appendix.

Short-range results are mixed, while aggregate gains and winning dataset–metric pairs increase with the horizon, mainly on flow datasets and absolute-error metrics.This horizon pattern is consistent with the model's division of labor. Recent observations provide a strong local signal at short horizons, while longer forecasts benefit from combining retained persistent-node states with dependencies reconstructed on the current graph. Context conditioning contributes an additional stage-level signal, but its effect remains dataset dependent, as reflected by the different gains across flow, air-quality, and energy measurements.

5.7 Temporal Forecasting and Latent-Context Similarity Analysis

We freeze the validation-best PEMS-Stream and JiNan models to examine node- and stage-specific prototype responses. Curves show 12th-horizon predictions for one new and three persistent nodes; heatmaps show responses to 20 prototypes. Full views appear in the appendix.

DyCSTA follows major PEMS-Stream cycles for new and persistent nodes. On JiNan, it retains the principal level and direction while smoothing some peaks. Prototype responses vary across nodes, with continuous weights combining related context patterns.

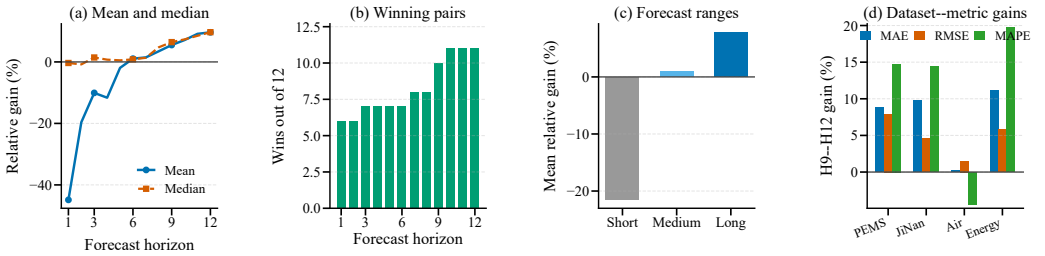

Figure 6: Forecast-range performance against fixed strong baselines: mean and median relative improvement, the number of winning dataset–metric pairs, range-group averages, and long-range behavior across all 12 horizons.

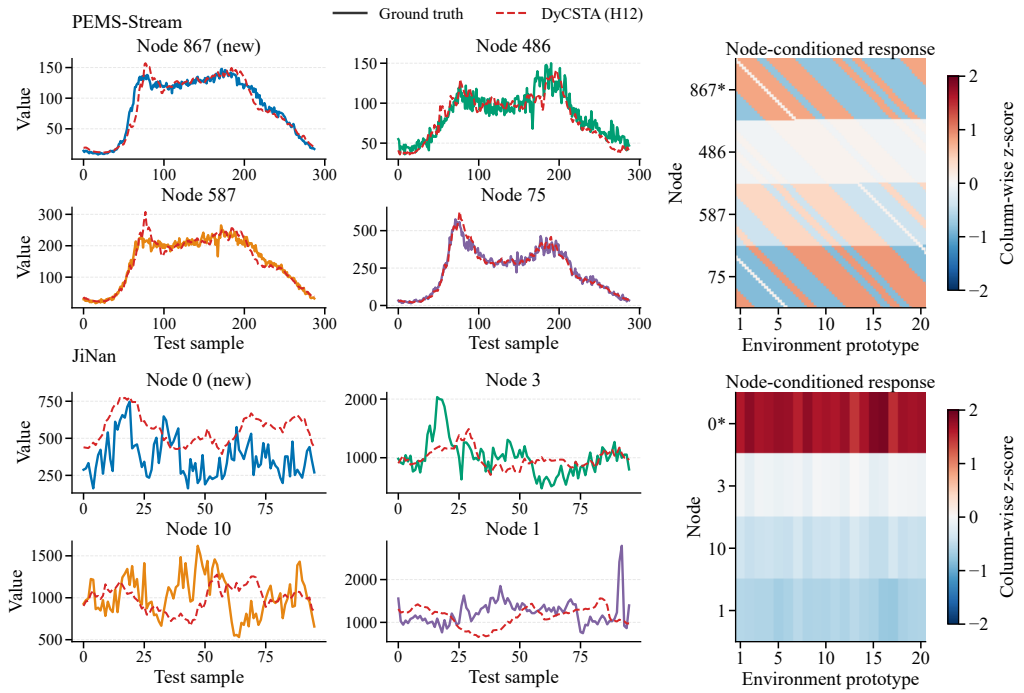

Figure 7: Representative forecasts and prototype responses on PEMS and JiNan. Dashed curves show DyCSTA; stars mark new nodes.

6 Conclusion

We presented DyCSTA, a model for multi-step forecasting on evolving graphs that combines persistent-node history retention, arriving-node modeling, current-graph dependency reconstruction, and latent-context-assisted prediction. The framework improves forecasting under changing node sets by aligning historical information with the current graph. However, its ability to capture evolving dependencies varies across graph scales and can be weaker at both very small and very large scales. Future work will study evolution modeling that better handles graphs of different sizes and forecasting mechanisms that transfer across changing topologies.

AI use statement

Generative AI tools assisted with language editing, LaTeX and figure-layout checks, and code–manuscript consistency review; the authors verified all scientific content, results, citations, and the final text.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815, 2020.

Namrata Banerji and Tanya Berger-Wolf. DynaSTy: A framework for spatiotemporal node attribute prediction in dynamic graphs. *arXiv preprint arXiv:2601.05391*, 2026.

Wei Chen and Yuxuan Liang. Expand and compress: Exploring tuning principles for continual spatio-temporal graph forecasting. In *International Conference on Learning Representations*, 2025.

Xu Chen, Junshan Wang, and Kunqing Xie. TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 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 AAAI Conference on Artificial Intelligence*, volume 37, pp. 4365–4373, 2023.

Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, 17(1):1–21, 2023.

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.

Aoyu Liu and Yaying Zhang. A general spatio-temporal backbone with scalable contextual pattern bank for urban continual forecasting. In *International Conference on Learning Representations*, 2026.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. STAEformer: Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4125–4129, 2023.

Minh Duc Nguyen and Viet Cuong Ta. Temporal structural preserving with subtree attention in dynamic graph transformers. *ACM Transactions on Knowledge Discovery from Data*, 19(4):1–24, 2025.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5363–5370, 2020.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. DySAT: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 519–527, 2020.

Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1696–1705, 2022.

Feihong Tan, Chunhui Zhang, and Lei Liu. DyAtGNN: Dynamic attention graph neural networks for dynamic graph. *Knowledge-Based Systems*, 325:113935, 2025.

Xiang Wu, Xunkai Li, Rong-Hua Li, Kangfei Zhao, and Guoren Wang. ScaDyG: A new paradigm for large-scale dynamic graph learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

Haoyu Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. STRAP: Spatio-temporal pattern retrieval for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 38:118006–118041, 2025a.

Hui Zhang, Kun Ding, Jietao Xie, Weidong Xiao, and Yuxiang Xie. Flow prediction via adaptive dynamic graph with spatio-temporal correlations. *Expert Systems with Applications*, 261:125474, 2025b.

Kexuan Zhang, Xiaobei Zou, and Yang Tang. Caformer: Rethinking time series analysis from causal perspective. *arXiv preprint arXiv:2403.08572*, 2024.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Zhaoqi Zhang, Miao Xie, Yi Li, Linyou Cai, Siqiang Luo, and Gao Cong. CausalPOI: Spatio-temporal graph-based causal modeling for cold-start POI check-in forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026. doi: 10.1145/3770855.3817641.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. GMAN: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1234–1241, 2020.

Jialun Zheng, Divya Saxena, and Jiannong Cao. COIN-GNN: Inductive spatial-temporal prediction for continuous distribution shifts via graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

A APPENDIX

This appendix provides additional material for the main paper, including dataset and protocol details, baseline settings, parameter sensitivity, complementary ablations, dynamic-node results, forecast-range diagnostics, and complete temporal forecasting cases. All results follow the same snapshot splits, active-node masks, prediction horizons, and evaluation metrics used in the main paper.

A.1 DATASET AND PROTOCOL DETAILS

The four datasets cover complementary evolving systems. Following the detailed organization used by EAC, we state the forecasting target, temporal partition, and graph evolution for each domain before reporting statistics. Table 2 reports the controlled JiNan snapshot transitions, and Table 3 lists the active nodes and nonzero directed adjacency entries supplied to every method. The counts describe the actual stage graphs used for training and evaluation rather than an abstract full graph.

PEMS-Stream. This benchmark represents traffic flow on a directed road-sensor network in Northern California. Each observation is sampled every five minutes, and the task predicts future sensor flow from the preceding 12 observations. Seven annual stages are constructed from the 31-day interval beginning July 10 in 2011–2017. The active set grows from 655 to 871 sensors and the directed edge set grows from 3,154 to 5,576 entries. A sensor absent from one stage but active in the next is treated as an arriving node; consequently, later stages jointly change graph size, connectivity, and traffic distribution instead of merely appending observations to a fixed network.

Air-Stream. Air-Stream describes hourly air-quality measurements collected by monitoring stations across China from 2016 to 2019. The prediction target is the future air-quality sequence at all active stations. Each calendar year forms one graph stage, giving four chronologically ordered stages and node growth from 1,087 to 1,202 stations. Following common spatio-temporal forecasting practice, geographic distance defines the stage-wise graph through a thresholded Gaussian affinity; a new monitor therefore changes both the node set and its spatial neighborhood. This domain evaluates the model beyond mobility flow on slower environmental dynamics and a substantially larger graph.

Energy-Stream. Energy-Stream is derived from the spatial dynamic wind-power forecasting data released for the KDD Cup benchmark. It contains ten-minute turbine measurements over 240 usable days; stage boundaries divide the sequence into four consecutive periods with 103, 113, 122, and 134 active turbines. The task predicts future turbine indicators from the preceding observations, using nacelle temperature as the forecasting signal in the aligned benchmark protocol. Spatial proximity constructs the evolving turbine graph, whose directed nonzero entries increase from 1,877 to 2,726. Compared with traffic and wireless flow, the observed wind-power series contains stronger local and exogenous variation, so the benefit of historical graph evolution can be weaker in this domain.

JiNan. JiNan contains real base-station traffic from September 3, 2021 to January 3, 2022 at 15-minute resolution. Because the raw measurements do not include a naturally logged deployment history, we retain the observed traffic values and simulate controlled activation and retirement of stations to form four monthly graph snapshots. This design isolates forecasting under explicit node arrival and departure while avoiding synthetic traffic values. Its detailed transition counts are reported in Table 2.

JiNan emulates the infrastructure changes motivating this work. Each monthly snapshot changes the active base-station set, and arrivals and departures in Table 2 are measured relative to the preceding snapshot. This construction creates a controlled evaluation on a held-out later-stage graph while preserving the temporal measurements of the real network.

Snap.	Date range	Nodes	Arr.	Dep.
1	Sep. 3–Oct. 2	8	–	–
2	Oct. 3–Nov. 2	9	3	2
3	Nov. 3–Dec. 2	10	2	1
4	Dec. 3–Jan. 3	11	1	0

Table 2: JiNan snapshot construction. Arrivals and departures are relative to the preceding snapshot.

Dataset	Domain	Time range	Stages	Node and edge evolution	Freq.	Frames
PEMS-Stream	Traffic	2011–2017; Jul. 10–Aug. 9	7	Nodes: $655 \to 715 \to 786 \to 822 \to 834 \to 850 \to 871$ Edges: $3154 \to 3858 \to 4632 \to 5072 \to 5188 \to 5382 \to 5576$	5 min	61,992
JiNan	Communication	Sep. 3, 2021–Jan. 3, 2022	4	Nodes: $8 \to 9 \to 10 \to 11$ Edges: $56 \to 72 \to 90 \to 110$	15 min	11,808
Air-Stream	Air quality	Jan. 1, 2016–Dec. 31, 2019	4	Nodes: $1087 \to 1154 \to 1193 \to 1202$ Edges: $3852 \to 4360 \to 4568 \to 4634$	1 h	34,065
Energy-Stream	Wind power	240 days	4	Nodes: $103 \to 113 \to 122 \to 134$ Edges: $1877 \to 2139 \to 2382 \to 2726$	10 min	34,560

Table 3: Dataset statistics. Node and edge sequences correspond to consecutive snapshots, and edge counts are nonzero directed entries in the adjacency matrices. PEMS-Stream contains the 31-day interval beginning July 10 in each year.

PEMS-Stream uses a chronological 6:2:2 split over the complete sequence. For JiNan, Air-Stream, and Energy-Stream, the first three snapshots provide training and validation data with a 7:3 split, and the final snapshot is reserved for testing. Evaluation therefore uses a future graph structure and distribution rather than a random subset of training snapshots. At every graph stage, each active node is encoded from the same P-step input window. An arriving node therefore uses its available P observations from the current stage after feature projection, without inheriting recurrent state from the preceding stage. For stability-aware selection, p_v^{t-1} and p_v^t are empirical distributions built from valid observations in adjacent stages, and W_1 denotes the first-order Wasserstein distance.

A.2 Baseline Details

The comparison set is organized to answer three distinct questions. **Dynamic spatio-temporal forecasters** test whether adapting spatial and temporal dependencies alone is sufficient: DMLSA learns adaptive dynamic graphs with local spatio-temporal correlations, while DynaSTy predicts node attributes under changing graph snapshots. **Dynamic-graph learners** test cross-snapshot structural modeling: DyAtGNN emphasizes influential nodes through dynamic attention, DySubTree preserves temporal structure with subtree attention, and ScaDyG targets scalable dynamic-graph transformations. **Continual and inductive forecasters** test resistance to evolving distributions: EAC expands and compresses model capacity, STRAP retrieves transferable spatio-temporal patterns, STBP maintains a scalable contextual pattern bank, and COIN-GNN addresses continuous shifts with inductive graph prediction. Full bibliographic entries for all baselines appear in the main paper.

All methods receive the same snapshot boundaries, active-node masks, chronological splits, 12-step input/output windows, and metric definitions. Checkpoints are selected only by validation performance. This grouping prevents the comparison from favoring one narrow family and clarifies whether the proposed design remains useful against dynamic forecasting, graph evolution, and continual adaptation baselines.

A.3 Hyperparameter Settings

Table 4 lists the shared training and model settings, while Table 5 gives the dataset-specific values. We omit implementation identifiers such as paths, device strings, and node counts. All checkpoints

Shared hyperparameter	Value
Batch size	32
Maximum epochs	100
Early-stopping patience	15
Learning rate	0.001
Weight decay	1×10^{-5}
Decay epoch	5
Dropout	0.05
Entmax parameter	2.0
Hidden dimension	64
Node embedding dimension	16
σ	0.9
GCN weight λ_{gcn}	0.5
GCN mixing α_{gcn}	0.3
Auxiliary coefficient decay	0.001
Random seed	42

Table 4: Shared training and model hyperparameters.

Dataset-specific hyperparameter	PEMS-Stream	JiNan	Air-Stream	Energy-Stream
Number of context groups K	8	4	4	8
Head dimension d	8	16	16	8
Environment prototypes	20	20	8	20
Leakage	0.9	0.8	0.8	0.9
Margin start	0.20	0.20	0.10	0.20
Margin end	0.05	0.05	0.02	0.05
α	0.05	0.05	0.02	0.05
β	0.01	0.01	0.06	0.01
Orthogonal weight	0.01	0.01	0.003	0.01
Orthogonal warmup	10	10	20	10

Table 5: Dataset-specific hyperparameters. K denotes the number of context groups used by the dynamic encoder; the environment-prototype count is the number of learnable prototype codes.

use the chronological validation protocol described above; the random seed is fixed at 42 for the reported runs.

A.4 Parameter Sensitivity

This section provides the complete sensitivity analysis referenced in the main paper. On JiNan, the environment-replacement weight α and prototype-usage weight β are independently varied over $\{0, 0.01, \ldots, 0.07\}$. When one changes, the other is fixed at its validation-selected value ($\alpha = 0.05$, $\beta = 0.01$). Figure 8 reports MAE, RMSE, and MAPE averaged over all 12 horizons, while Figure 9 presents the horizon-wise curves.

The aggregate errors fluctuate across settings, but neighboring values do not cause consistent or substantial degradation. Different metrics favor slightly different weights, yet the horizon-wise curves retain similar error-growth patterns. Thus, the reported behavior is not tied to one isolated parameter combination.

Figure 9 tests whether a favorable average is caused by only one or two horizons. Across MAE, neighboring settings produce similarly shaped curves and preserve gradual error accumulation. RMSE is less uniform because different weights trade typical errors against a small number of large deviations, while MAPE is more sensitive when targets are small. Together, the results show a usable parameter region rather than dependence on a numerically precise value.

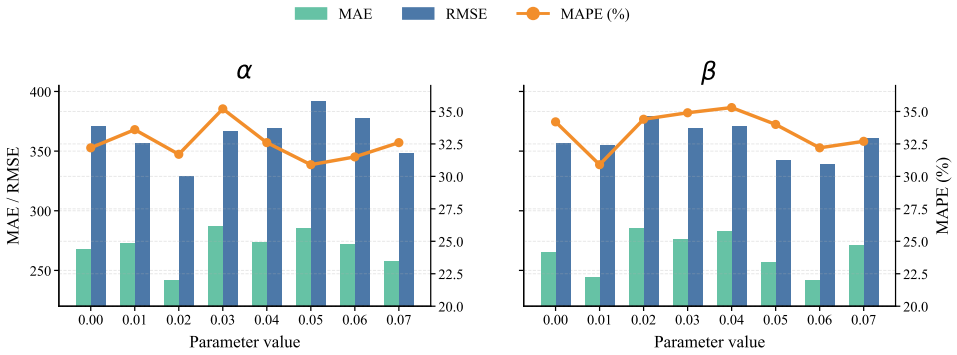

Figure 8: Overall JiNan sensitivity to α (left) and β (right). Bars show MAE/RMSE and lines show MAPE.

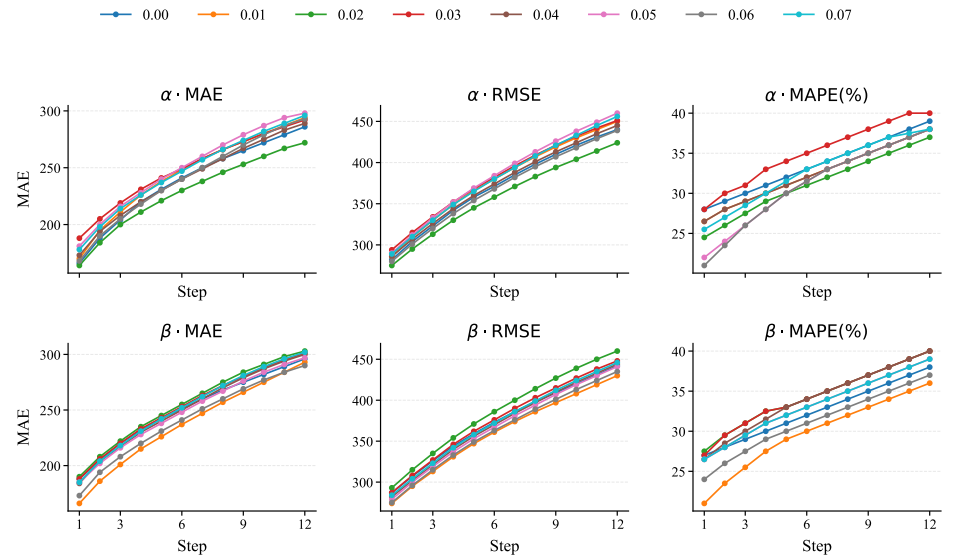

Figure 9: Horizon-wise JiNan sensitivity while varying α (top row) and β (bottom row). Columns report MAE, RMSE, and MAPE with one shared parameter-value legend.

A.5 Complete Ablation Metrics

The main paper reports the MAE ablation, while Figures 10 and 11 provide the complementary RMSE and MAPE results for the same seven variants. DyCSTA obtains the lowest average RMSE on PEMS-Stream, JiNan, and Energy-Stream. The MAPE results also favor DyCSTA on PEMS-Stream and Energy-Stream. JiNan is an exception: NO-DYN and NO-ENV obtain lower average MAPE than the full model, indicating that the small JiNan graph produces metric-dependent differences in the auxiliary variants.

Across the three metrics, the spatio-temporal ablations consistently increase the error relative to DyCSTA. NO-TCA is particularly detrimental on JiNan, while NO-GCF produces a larger degradation on PEMS-Stream. On Energy-Stream, removing TCA causes the largest increase among the spatio-temporal variants, whereas removing dynamic evolution or the causality-inspired branch also increases the error relative to the full model.

The three datasets expose different amounts of graph evolution. PEMS-Stream grows from 655 to 871 active sensors, JiNan grows from 8 to 11 active base stations, and Energy-Stream grows from 103 to 134 active turbines. The weaker separation between DyCSTA and NO-DYN on JiNan is therefore consistent with its smaller active-node set and fewer available transitions. This result does

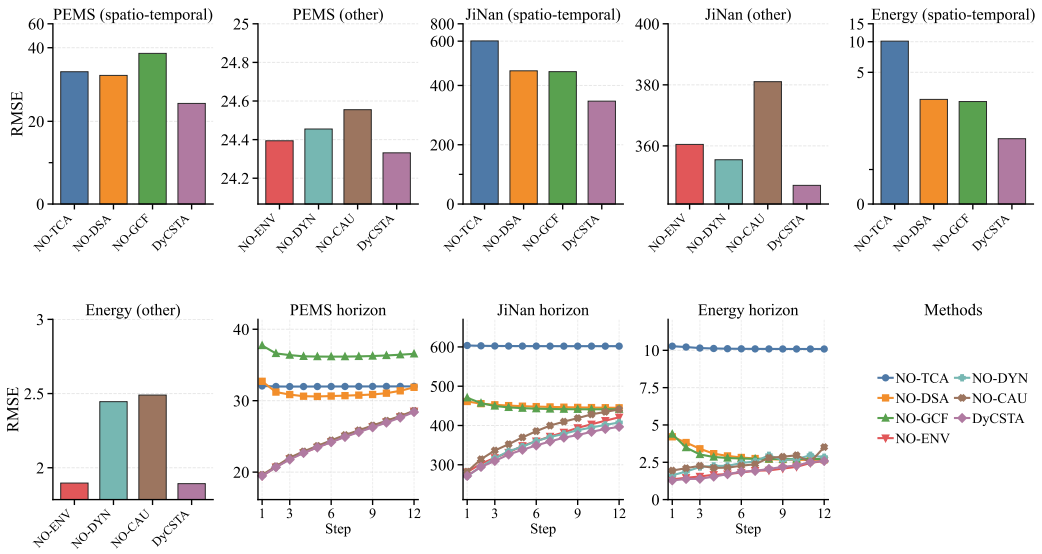

Figure 10: RMSE ablations for seven variants on PEMS-Stream, JiNan, and Energy-Stream. Bars show the 12-step average and line panels show horizon-wise RMSE.

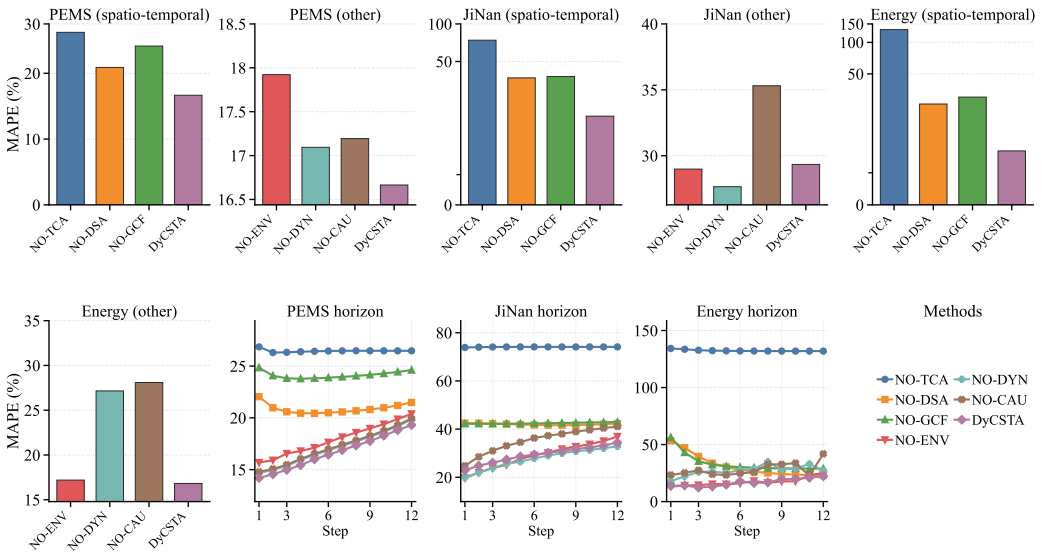

Figure 11: MAPE ablations for seven variants on PEMS-Stream, JiNan, and Energy-Stream. Bars show the 12-step average and line panels show horizon-wise MAPE in percent.

not imply that dynamic evolution is unnecessary; it indicates that its predictive contribution is less visible when the evolving graph contains only a small number of nodes.

A.6 Predictive Representation Separation and Environment Assistance Details

The main paper reports the complete matched-probe comparison among the forecast-relevant predictor C_0, the context-assisted branch C_E, and the context branch T_E; it is not repeated here. Both probes use the same frozen encoder and identically initialized decoders, so their MAE, RMSE, and MAPE ratios compare the predictive utility of the two learned representations under matched probes. The C_E/C_0 curves are normalized independently at every horizon; the redundant constant $C_0 = 100\%$ reference is omitted.

All three T_E/C_E ratios substantially exceed one, indicating that the matched probe obtains lower forecasting error from the forecast-relevant branch than from the context branch. Context-assisted C_E reduces MAE and RMSE at every horizon relative to C_0 and improves MAPE on average, although the latter fluctuates around the reference at individual steps. Within this matched-probe analysis, the gain is more consistent for absolute-error metrics than for MAPE, and the prototypes function as latent context codes within the forecasting model.

This comparison controls two common alternative explanations. First, C_E and T_E are evaluated with identically initialized probe decoders, so the ratio cannot be attributed to one branch receiving a stronger predictor. Second, the encoder is frozen during probing, preventing the representation from adapting to the diagnostic objective. The relative curves compare C_E and C_0 at the same horizon, which also prevents natural horizon-wise error growth from being mistaken for an environment gain.

A.7 Additional Dynamic-Node Results

The main paper presents MAE for a preselected group of strong baselines to keep the double-column figure readable. Figure 12 restores all evaluated methods and reports MAE, RMSE, and MAPE for all, existing, and newly arriving PEMS-Stream nodes. The node groups, checkpoints, and 12 horizons are unchanged.

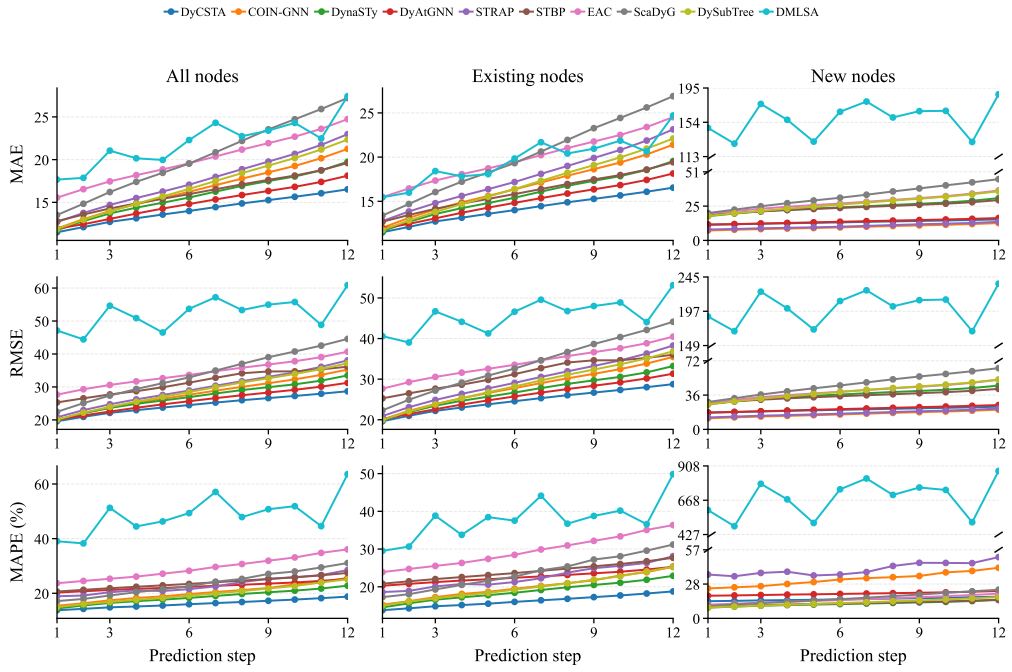

Figure 12: Dynamic-node MAE, RMSE, and MAPE for all evaluated methods on all, existing, and newly arriving PEMS-Stream nodes. The three metric rows use linear axes with plain-number tick labels; the marked interval in the new-node column indicates the compressed range.

Across all and existing nodes, DyCSTAkeeps the lowest or near-lowest errors across the three metrics, so the main-paper MAE trend is not an artifact of showing fewer baselines. Newly arriving nodes remain harder, and MAPE is not lowest at every horizon; nevertheless, the full comparison shows competitive predictive behavior when previously unseen entities enter the graph.

A.8 Forecast-Range Performance

The main table can conceal how uncertainty accumulates over the complete prediction window. For every dataset–metric pair, one strong baseline is selected once by its 12-step average and held fixed at

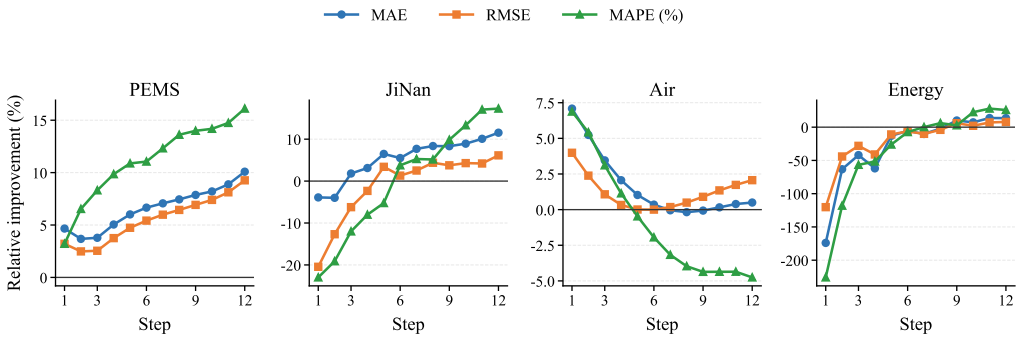

Figure 13: Dataset-specific forecast-range improvements against fixed baselines. The four subplots show PEMS-Stream, JiNan, Air-Stream, and Energy-Stream in a single row.

all horizons. Relative improvement is $(e_b - e_o)/e_b$, where e_b and e_o are baseline and DyCSTAerrors. Figure 13 reports the dataset-specific 12-step curves rather than repeating the compact summary used in the main paper.

PEMS-Stream is positive throughout and improves steadily with range. JiNan moves from an early disadvantage to positive gains in all metrics. Air-Stream crosses toward positive MAE/RMSE at later steps, although early percentages are magnified by the baseline's small absolute errors and MAPE is not uniformly better. Energy-Stream stays close to DyAtGNN: MAE and RMSE recover by the final horizons while MAPE remains lower. The different domain trends indicate that the benefit of historical graph evolution and context conditioning varies across datasets. The aggregate long-range result therefore reflects strong flow-domain gains and improving absolute errors, not uniform superiority on every domain and metric. The four single-column panels intentionally separate datasets so that visually larger percentage gains from one domain do not compress the remaining curves. Values above zero mean that DyCSTA has lower error than the fixed comparator; values below zero expose the horizons where the comparator remains stronger. The plots therefore complement, rather than replace, the absolute errors in the main table.

The fixed-comparator protocol is essential for interpreting this trend. Selecting a different baseline at each horizon could manufacture an apparent improvement by changing the reference method. Here, each dataset–metric pair uses one baseline chosen once from its 12-step average. Consequently, a rising curve means that the same comparison becomes more favorable as uncertainty accumulates. Reporting both mean and median further limits the influence of one unusually large dataset–metric gain.

The range curves also motivate the complete temporal cases below. The aggregate plots show where long-horizon gains appear, but they do not reveal whether the gains come from stable periodic tracking, from smoother behavior on persistent nodes, or from retaining usable predictions for newly arriving nodes. The case studies therefore inspect H12 predictions and prototype responses together, using the same frozen checkpoints and without selecting nodes by prediction error.

Across datasets, the curve shapes also show different sources of difficulty. PEMS-Stream has a consistent upward trend, which means the fixed comparator becomes relatively weaker as the horizon grows. JiNan is less favorable at the first few horizons but turns positive later, suggesting that the model is more useful when short-term local noise gives way to broader temporal structure. Air-Stream improves mainly in MAE/RMSE at later horizons, while Energy-Stream remains closer to the comparator because its observed series contains stronger local and exogenous variation that is not represented directly in the graph. These distinctions are why the supplementary cases emphasize both prediction traces and latent-response heatmaps instead of relying only on aggregate improvement curves.

A.9 Temporal Forecasting and Latent-Context Similarity

Purpose and protocol. We freeze the validation-best PEMS-Stream and JiNan checkpoints and perform inference-only diagnostics without retraining. Each overview point is the 12th-horizon output

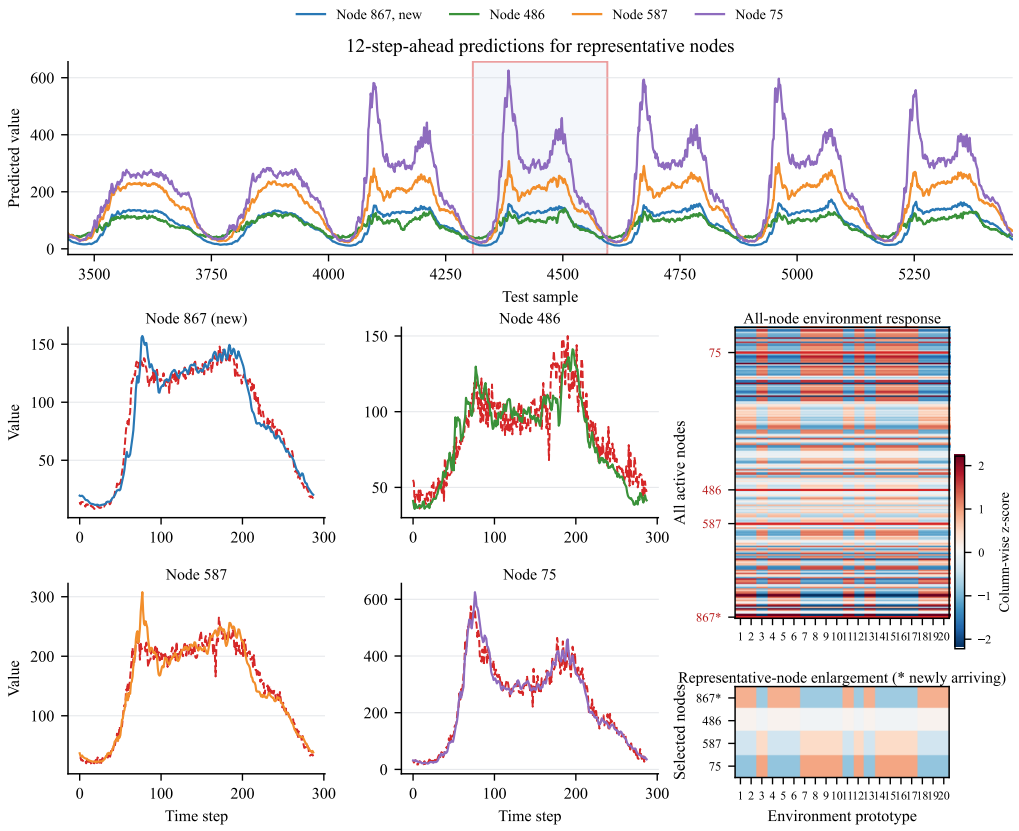

Figure 14: Complete PEMS-Stream temporal forecasting and latent-context case. The upper plot shows H12 predictions over the complete displayed test sequence and marks the detailed interval in red. The lower-left plots compare ground truth and predictions separately for four representative sensors. The lower-right panels show all-node context responses and an enlargement of those four sensors in the final snapshot.

of one rolling input window; the red rectangle locates the detailed ground-truth interval shown below. One newly arriving and three persistent nodes are selected by activity, scale, and volatility on a disjoint context interval without using prediction errors. Context heatmaps report column-standardized responses of the context representation to 20 prototypes before weighted fusion: rows test node dependence and columns test prototype selectivity. Colors show relative prototype responses rather than accuracy values or hard environment labels.

PEMS-Stream case. The main paper reports a compact cross-dataset view to show prediction behavior and latent-context responses jointly. The supplementary case enlarges that view and examines whether the local interval is representative of the longer test sequence, whether both new and persistent sensors retain H12 fidelity, and whether the context representation remains differentiated when all active sensors are examined. The four nodes are selected without prediction-error information and span different traffic scales; the asterisk denotes the newly arriving sensor.

Prediction analysis. The upper overview contains repeated daily cycles and peak–decline transitions, and the red interval lies inside a recurring regime rather than a unique low-variance segment. The model tracks sustained levels, peak locations, and descending trends for both the newly arriving sensor and persistent sensors; errors become more visible around abrupt narrow peaks, consistent with uncertainty accumulated by the 12th horizon.

Context analysis. The all-node heatmap supplies a population-level check beyond the four displayed sensors. Recurring vertical bands indicate shared context, whereas row-wise variation shows that the response is not a single graph-wide code. The newly arriving sensor follows the main rise–fall pattern without inheriting a preceding-stage state, while prototype responses are not ordered simply

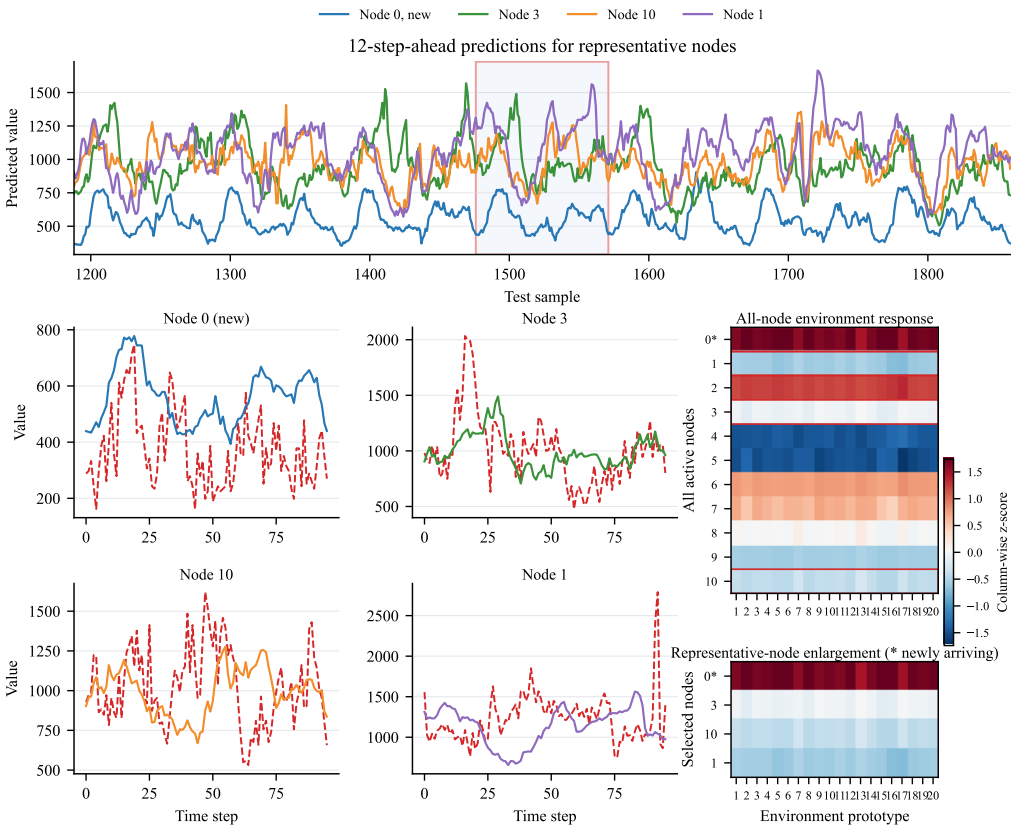

Figure 15: Complete JiNan temporal forecasting and latent-context case. The upper plot gives H12 predictions over the displayed test sequence and highlights the detailed interval. The lower-left plots overlay ground truth and predictions for four representative stations. The lower-right panels report final-snapshot prototype responses for all active stations and enlarge the same four representatives.

by node age. This supports using the context representation as contextual evidence rather than as a proxy for whether a node is new or persistent.

The PEMS-Stream case separates effects that are merged in aggregate MAE/RMSE curves. Persistent sensors preserve sharper daily cycles because their historical dynamics also contribute to the evolved state, while the new sensor relies on the current-stage input window and graph propagation. The response heatmap shows that both types of nodes can share some prototypes while still keeping distinct response profiles.

The displayed interval also clarifies how the model behaves around abrupt traffic changes. The strongest deviations occur near narrow peaks, where the 12-step horizon requires anticipating a quick rise or decay from earlier observations. In smoother intervals, the predicted curves retain both amplitude and phase for the selected sensors. This pattern is consistent with the main-paper horizon analysis: long-range gains are strongest when the model can exploit repeated temporal structure, but very sharp local events remain the main source of residual error.

For the prototype responses, the important observation is not that a particular prototype has a fixed semantic label. Instead, different nodes activate different mixtures before environment fusion. The new node does not collapse into an average profile, and persistent nodes do not share identical responses. This supports the use of a soft latent condition, because it lets related nodes share contextual evidence while preserving node-level heterogeneity after topology changes.

JiNan case. JiNan complements the regular road-traffic setting with weaker periodicity, sharper fluctuations, and explicit base-station arrival and departure.

Prediction analysis. We use the same frozen-checkpoint protocol and the same selection rule for one newly arriving and three persistent stations, which controls the analysis across datasets while testing a smaller, more irregular dynamic graph. The complete sequence shows sustained but non-identical regimes and places the detailed interval within repeated rises and drops. The overlays retain broad levels and directions across different scales, including the lower range of the newly arriving station. Some short high-frequency peaks are smoothed or shifted, reflecting the difficulty of H12 forecasting on irregular wireless traffic and sharper local demand changes.

Environment analysis. All 11 active stations exhibit row-level contrasts, and columns vary in intensity across stations. The newly arriving station differs from persistent stations, while persistent stations span positive, near-neutral, and negative prototype groups. The all-node panel verifies that this is not an artifact of selecting four examples. These responses are consistent with C retaining the principal predictive pathway and T supplying differentiated context, without assigning fixed physical meanings to individual prototypes.

Compared with PEMS-Stream, JiNan contains fewer active nodes and less regular periodicity, so each station's local fluctuation has a larger visual effect on H12 prediction. The model still preserves broad levels and directions for both new and persistent stations, but the overlays make clear where sharp peaks are smoothed or shifted. This case therefore complements the aggregate metrics by exposing the main remaining uncertainty rather than only showing average improvement.

The JiNan heatmap is intentionally shown for all active stations because a four-node example alone could overstate heterogeneity. The all-node panel shows row-level differences across the full final snapshot, while the representative enlargement makes the selected stations readable. The newly arriving station has a response pattern that differs from persistent stations, but persistent stations also vary among themselves. This indicates that the latent response is not merely a node-arrival indicator.

Taken together, the PEMS-Stream and JiNan cases show two complementary regimes. PEMS-Stream demonstrates regular high-volume traffic with clear daily structure, where topology evolution and temporal periodicity are both helpful. JiNan demonstrates a smaller and noisier communication graph, where predictions retain broad movement but smooth some sharp events. The qualitative evidence is therefore aligned with the quantitative results: the method is strongest when graph evolution and temporal regularity provide reusable structure, while irregular local spikes remain more difficult.