REACT: A Lightweight Reliability-Aware Framework for Spatio-Temporal Out-of-Distribution Prediction

Yongfeng Su[1] **Ziquan Fang**[1] **Jihua Yang**[1] **Wei Shao**[1] **Yunjun Gao**[2]
[1]School of Software Technology, Zhejiang University
[2]College of Computer Science and Technology, Zhejiang University
{yfsu, zqfang, jihuayang, wei.shao, gaoyj}@zju.edu.cn

Abstract

In the real world, spatio-temporal forecasting systems inevitably encounter out-of-distribution (OOD) shifts, where temporal shifts evolve due to seasonal patterns or policy changes, and spatial dependencies shift with sensor deployment, removal, or topology reconfiguration. While many approaches attempt to address these challenges via invariant learning or causal adjustment, they implicitly assume that the contextual signals supporting prediction remain reliable. Through empirical studies and theoretical analysis, we find that this assumption often breaks because: (i) node-level context can become biased or incomplete under temporal and spatial OOD, yet existing methods lack mechanisms to assess such reliability; (ii) existing standard graph-based aggregation propagates information with fixed strength, potentially amplifying corrupted context across nodes. To this end, we propose **REACT**, a lightweight **RE**liability-**A**ware **C**ontext-calibrated spatio-**T**emporal forecaster that explicitly models and controls context reliability under OOD. REACT decouples prediction from adaptation by first establishing a topology-independent predictive anchor via a graph-free linear encoder, ensuring transferable node-wise representations. It then constructs an availability-aware context that combines observation statistics with static priors and an explicit prior-availability indicator, enabling explicit characterization of context completeness. Building on this, REACT introduces a reliability-guided calibration mechanism that dynamically modulates spatial support through uncertainty-aware gating, so that each node draws on its neighbors according to its own context uncertainty. Extensive experiments on ten real-world datasets with diverse temporal and structural shifts demonstrate that REACT consistently outperforms eleven baselines. These results highlight the importance of explicitly modeling and controlling context reliability as a fundamental principle for robust spatio-temporal forecasting under distribution shift.

1 Introduction

Spatio-temporal prediction aims to forecast the future state of dynamic systems from historical spatio-temporal observations and underpins many real-world applications, such as traffic flow forecasting [17, 38] and crowd flow forecasting [41, 42]. Recent Spatio-Temporal Graph Neural Networks (**STGNNs**) achieve strong performance by coupling spatial neural operators with temporal dynamics modeling [2, 12, 17, 34, 38, 43]. However, most existing methods are developed under an **In-Distribution** assumption, where ***training and test data share similar*** *temporal regimes, node semantics, and graph structures* [20, 28, 38].

Real-world deployment violates this assumption. In practice, spatio-temporal systems frequently encounter **out-of-distribution (OOD) shifts** [23, 31, 36, 45]. In traffic networks, temporal shifts arise when evolving environments—such as policy changes or seasonal transitions—alter demand

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

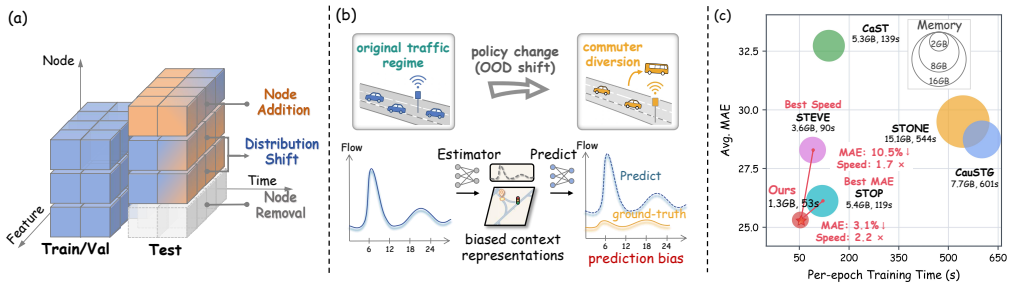

Figure 1: (a) Temporal and structural OOD shifts, including distribution changes, node addition, and removal; (b) Illustration of context misestimation under OOD; (c) Accuracy–efficiency trade-off on GBA, where our method (i.e., the red bubble) performs SOTA (bubble size: peak GPU memory).

dynamics and invalidate learned temporal regularities. Spatial shifts occur when sensors are added or removed, changing graph topology and introducing unseen nodes without training-time history. These shifts often co-occur, jointly affecting both temporal dynamics and spatial dependencies, and thus give rise to a challenging spatio-temporal OOD setting, as illustrated in Fig. 1(a).

Recent approaches attempt to address OOD through invariant learning [31, 33, 45], causal adjustment [10, 36], or environment augmentation [16]. Despite their differences, many of them still follow a *learn-then-trust* paradigm: representations or contextual signals learned from source environments are trusted at deployment. This paradigm improves training-time robustness, but *leaves a reliability gap because it does not explicitly assess whether node-level context remains reliable under deployment shifts.* As formalized in **Appendix B** (Corollary 1), unconditional context trust incurs positive excess or irreducible context risk when temporal shifts change the evidence-to-context mapping or spatial shifts remove context-relevant information. This gap motivates the first challenge: context can be misestimated before any downstream prediction or graph propagation is applied.

Challenge 1: Context misestimation under OOD shifts. Context is essential for adapting predictions to new environments, yet it can itself be distorted under OOD shifts. As shown in Fig. 1(b), context refers to the node-level operating state inferred from observation statistics, learned features, static priors, or spatial information. Consider a highway sensor trained under a stable morning-peak regime. After a policy change diverts commuters to public transit, its observations may exhibit flatter peaks and lower average flow. A context estimator trained on the original regime still maps these statistics through the same learned function, but their semantics have changed, leading to biased context representations. For newly deployed sensors, the problem is more severe due to limited observations and incomplete priors. The core issue is not a malfunction of individual modules, but that the model cannot determine when its inferred context has deviated from the training regime. **Appendix B.1** formally shows that under temporal OOD, the optimal context estimator is environment-dependent, and applying a source-trained estimator in the target environment incurs strictly positive excess context risk. **Appendix B.2** further shows that spatial OOD degrades the available evidence for context inference, introducing irreducible estimation error. ***Together, these results show that context misestimation is a structural risk under OOD, rather than a merely empirical failure case.***

Challenge 2: Uncontrolled propagation of unreliable context. A natural approach for unreliable local context is to leverage graph aggregation. However, existing widely used fixed spatial propagation cannot adapt to node-level context uncertainty. Some nodes should rely primarily on their own observations: for example, a stable residential sensor may be harmed if it blindly mixes signals from shifted nearby roads. In contrast, newly added sensors may benefit from increased neighborhood support to stabilize their uncertain context. Standard STGNNs treat these cases similarly because aggregation strength is fixed by the architecture or learned globally, rather than determined by the target node's current context quality. As a result, unreliable context is not only unrecognized but also propagated across the graph, potentially amplifying estimation errors. The analysis in **Appendix B.3** formalizes this effect by showing that the calibrated context error decomposes into local estimation error, propagated neighbor error, and neighborhood mismatch. ***Without node-specific control of propagation strength, these error terms can accumulate rather than cancel out.***

Contributions. To address these challenges, we propose **REACT**, a lightweight **RE**liability-**A**ware **C**ontext-calibrated spatio-**T**emporal framework that explicitly models and controls context reliability. As shown in Fig. 1(c), REACT consistently improves upon state-of-the-art baselines, reducing Avg. MAE by up to 10.5%, while achieving up to 2.2× faster training and 4.3× lower memory usage.

- **Reliability-aware OOD Prediction.** We identify *context reliability* as a fundamental bottleneck. We provide empirical and theoretical analysis showing distribution shifts induce hidden bias in local context or remove context-relevant information, leading to unavoidable estimation risk and uncontrolled error propagation in existing methods. Motivated by this, we propose a decoupled framework to separate prediction from context estimation and reliability-aware calibration.
- **Context Reliability Modeling.** We design the *Availability-Aware Context Estimator* to construct local context from observation statistics and static priors, and estimate whether contextual information remains trustworthy under temporal and structural distribution shifts.
- **Adaptive Context Calibration.** We introduce *Reliability-Guided Context Calibration* and *Gated Context Modulation* to dynamically adjust contextual representations according to their estimated reliability and inject calibrated context into the predictive anchor.
- **Extensive Evaluation.** Experiments on ten real-world datasets with diverse temporal and structural shifts demonstrate that REACT achieves state-of-the-art OOD performance across eleven baselines in terms of accuracy, efficiency, and GPU usage.

2 Related Work

Spatio-Temporal Prediction. Spatio-temporal forecasting has evolved from classical statistical models [7, 9, 22] to deep neural architectures that more effectively capture temporal and spatial dependencies. Early deep models rely on recurrent and convolutional structures to model sequential dynamics and localized patterns [5, 6, 14, 29, 41, 42]. More recent approaches adopt graph neural networks, adaptive spatial operators, and heterogeneity-aware designs to encode non-Euclidean interactions and long-range dependencies, achieving strong in-distribution performance [2, 8, 15, 28, 30, 34]. Another line of work improves transferability through pre-training or universal spatio-temporal representations [18, 39]. *However, most existing predictors assume a fixed graph structure and relatively stable temporal regimes, leading to predictive operators and latent representations that are tightly coupled to the training distribution. This coupling makes them brittle under structural changes or unseen temporal dynamics at deployment.* In contrast, REACT decouples prediction from spatial interaction by establishing a topology-independent predictive anchor, while reserving graph interaction for context calibration under uncertainty, thereby improving robustness to both temporal and structural shifts.

Out-of-Distribution Learning. Out-of-distribution (OOD) learning aims to generalize under distribution shifts by discovering invariant relations, modeling causal mechanisms, or learning robust prediction rules [1, 4, 11, 26, 44]. In spatio-temporal forecasting, CaST [36] performs causal adjustment, and CauSTG [45] captures invariant relations across environments. STONE [31] addresses both temporal and structural shifts by constructing semantic graphs and learning invariant dependencies through environment perturbation. STEVE [16] learns confounder-aware representations via spatial-temporal self-supervision, while STOP [23] replaces direct message passing with centralized interactions and introduces perturbations to improve robustness. *Despite their differences, these methods share a common assumption: the learned representations or contextual signals remain reliable at test time. As a result, they focus on improving invariance or robustness during training, but do not explicitly assess the reliability of context under deployment shifts. Moreover, spatial interaction is typically applied uniformly, without considering node-specific uncertainty.* REACT is complementary to these approaches. Instead of relying solely on training-time robustness, it explicitly estimates context reliability at inference time and adaptively calibrates spatial interaction, enabling selective propagation under OOD conditions. Retrieval-augmented and continual methods, such as STRAP [40], STEV [24], EAC [3], and STBP [19], instead exploit data from newly arriving periods, either by enriching a retrieval library or by retraining or incrementally tuning model components; **Appendix E.1** contrasts these paradigms with the source-only setting of REACT.

3 Preliminary

The commonly used notations and descriptions are summarized in **Appendix A**.

Definition 1 (Spatio-Temporal Data). Spatio-temporal data consists of node observations evolving over time on a spatial graph. Formally, the graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ is the node set with $N = |\mathcal{V}|$ nodes, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix encoding spatial dependencies. Let $\mathbf{X}_t \in \mathbb{R}^{N \times d}$ denote the node features at time t. Given the past T time steps, the historical

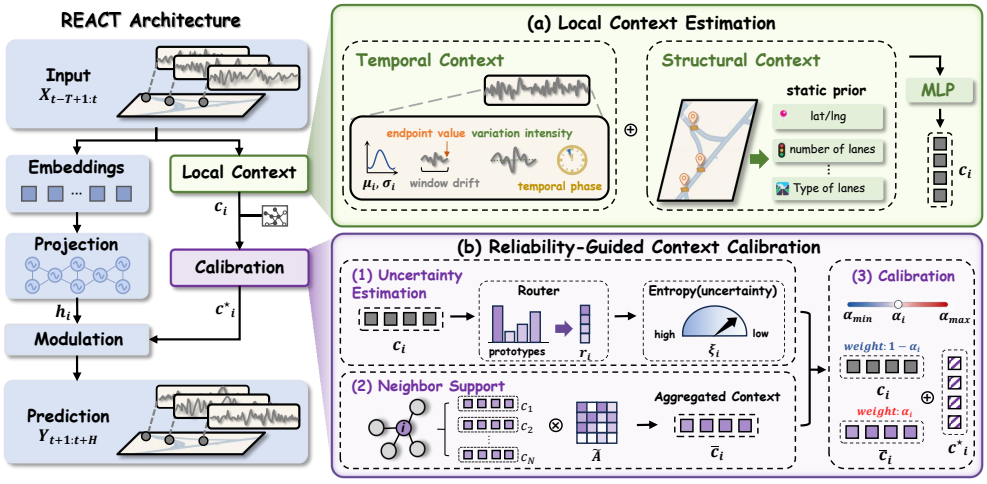

Figure 2: An Overview of REACT.

observations are

$$\mathbf{X} = [\mathbf{X}_{t-T+1}, \ldots, \mathbf{X}_t] \in \mathbb{R}^{T \times N \times d}.$$

Definition 2 (Spatio-Temporal OOD Prediction). In the out-of-distribution (OOD) setting, data are generated from multiple environments $e \in \mathcal{S}$, each characterized by a spatial graph $\mathcal{G}^e = (\mathcal{V}^e, \mathbf{A}^e)$ and a joint data distribution $p^e(\mathcal{X}, \mathcal{Y})$. OOD shifts may arise from both *temporal changes* in data distributions and *structural changes* in graph topology, such as node addition or removal.

Given historical observations

$$\mathbf{X} = [\mathbf{X}_{t-T+1}, \ldots, \mathbf{X}_t] \in \mathbb{R}^{T \times N^e \times d},$$

the goal is to learn a predictor f from a training environment e^* that generalizes to unseen environments $e \sim \mathcal{S}$, where both the data distribution and graph structure may differ.

The objective is formulated as

$$\min_f \sup_{e \in \mathcal{S}} \mathbb{E}_{(\mathbf{X},\mathbf{Y}) \sim p^e} \left[\mathcal{L}\big(f(\mathbf{X}, \mathcal{G}^e), \mathbf{Y}\big) \right], \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{H \times N^e}$ denotes the corresponding H-step future target matrix.

4 Methodology

Framework Overview. As illustrated in Fig. 2(left), REACT is built on a simple yet critical principle: *prediction should rely on transferable node-local dynamics, while adaptation should depend on explicitly calibrated context*. This separation is essential under OOD shifts, where context may become unreliable and should not be directly entangled with prediction.

Accordingly, REACT decouples prediction from adaptation through four coordinated modules. First, a *graph-free linear encoder* extracts a node-wise predictive anchor from historical observations and calendar cues, capturing transferable local dynamics independent of graph structure. Second, an *availability-aware context estimator* summarizes the node state from observations and optional priors, explicitly modeling context completeness. Third, a *reliability-guided calibration* module refines uncertain context by incorporating neighbor information with a strength set by each node's estimated uncertainty. Finally, a *gated context modulation* mechanism uses the calibrated context to adjust the predictive anchor before decoding adaptively. Next, we detail these modules in order.

4.1 Graph-Free Linear Encoder

Node-local Temporal Embedding. To encode node-local temporal information without graph propagation, REACT combines a full-window sequence embedding with calendar embeddings. Given the historical window $\mathbf{X} \in \mathbb{R}^{T \times N \times d}$, we flatten the local window of node i and project it as

$$\mathbf{x}_i^{\text{win}} = \text{vec}(\mathbf{X}_{:,i,:}) \in \mathbb{R}^{Td}, \qquad \mathbf{e}_i^{\text{seq}} = \phi_s(\mathbf{x}_i^{\text{win}}) \in \mathbb{R}^{d_s}. \tag{2}$$

Here, ϕ_s is a shared 1×1 convolution, equivalent to a node-wise linear projection over temporal-feature channels. We then concatenate the sequence embedding with learnable time-of-day (TOD) and day-of-week (DOW) embeddings, where τ_i and δ_i are the TOD and DOW indices of the last observed step:

$$\mathbf{z}_i^{(0)} = \left[\mathbf{e}_i^{\text{seq}},\ \mathbf{e}_{\tau_i}^{\text{tod}},\ \mathbf{e}_{\delta_i}^{\text{dow}}\right] \in \mathbb{R}^{d_h}. \tag{3}$$

The resulting $\mathbf{z}_i^{(0)}$ is the initial spatio-temporal embedding passed to the linear backbone. It contains no node identity or adjacency-dependent feature, keeping the predictive anchor graph-free and node-local.

Residual Projection Backbone. The node-local temporal embedding is then processed by L residual projection blocks shared across all nodes. Each block uses two shared 1×1 channel projections, equivalent to node-wise linear layers, with a ReLU activation in between:

$$\mathbf{z}_i^{(l+1)} = \mathbf{z}_i^{(l)} + \phi_2^{(l)}\Big(\text{ReLU}\Big(\phi_1^{(l)}(\mathbf{z}_i^{(l)})\Big)\Big), \tag{4}$$

where $\phi_1^{(l)}$ and $\phi_2^{(l)}$ are shared 1×1 convolutional channel-mixing projections. The final graph-free predictive anchor is

$$\mathbf{h}_i = \mathbf{z}_i^{(L)} \in \mathbb{R}^{d_h}. \tag{5}$$

Because the same residual projection block is applied independently to every node, this branch performs purely node-wise channel mixing with no message passing. $\mathbf{h}_i$ therefore represents how node i is expected to evolve before any structural correction is introduced. The ReLU only adds lightweight local nonlinearity, while the encoder remains graph-free and free of node-identity or adjacency-dependent parameters.

4.2 Availability-Aware Context Estimator

Observation Context. To address context misestimation under OOD shifts, REACT constructs local context explicitly from observation statistics rather than deriving it from predictive hidden states, as shown in Fig. 2(a). This separation avoids forcing a single representation to simultaneously support prediction and adaptation, which becomes unreliable when distributional semantics change.

The context branch aims to capture the node's *operating regime* rather than improve short-term prediction. Under spatio-temporal shifts, this regime may change along multiple axes, including level, variability, trend, and temporal phase. We summarize these factors using a compact regime descriptor:

$$\boldsymbol{\omega}_i = \left[\,\mu_i,\ \sigma_i,\ s_i^{(T)},\ s_i^{(T)} - s_i^{(1)},\ \bar{v}_i,\ \tau_i/N_T\,\right], \tag{6}$$

where μ_i and σ_i are the mean and standard deviation of the target history $\mathbf{s}_i = (s_i^{(1)}, \ldots, s_i^{(T)})$, and $\bar{v}_i = \frac{1}{T-1}\sum_{t=1}^{T-1} |s_i^{(t+1)} - s_i^{(t)}|$ denotes the mean absolute first-difference. These statistics provide a structured summary of distributional characteristics, capturing shifts in marginal scale, temporal dynamics, and periodic phase.

Unlike the spatio-temporal embedding path, which learns from the full historical window to support short-term forecasting, the context descriptor intentionally uses explicit window-level statistics and temporal phase. Its role is to identify regime changes and context reliability rather than maximize predictive capacity.

Availability-Aware Prior Fusion. Let $\mathbf{p}_i$ denote the static prior of node i, constructed from metadata and/or structural cues and normalized over training nodes. To account for incomplete or missing auxiliary information, REACT introduces an explicit availability mechanism. During training, priors are randomly masked at the node level, and a binary indicator $m_i \in \{0, 1\}$ is used to denote whether the prior is available. The local context is then estimated as

$$\mathbf{c}_i = \text{LN}\Big(\phi_c\big([\boldsymbol{\omega}_i,\ m_i\mathbf{p}_i,\ m_i]\big)\Big). \tag{7}$$

Here, ϕ_c denotes the learnable context projection network. The availability indicator is critical: a zero-valued prior is not equivalent to an unavailable one. By explicitly modeling m_i, REACT distinguishes between genuine prior information and missing inputs, enabling robust context estimation under partial or unreliable auxiliary data.

4.3 Reliability-Guided Context Calibration

Uncertainty Estimation. Under OOD shifts, the local context may become unreliable due to regime changes or incomplete priors. However, directly propagating neighbor information can introduce additional noise from shifted nodes. To address this trade-off, REACT explicitly estimates the reliability of local context before applying spatial interaction, which is shown in Fig. 2(b).

Given the raw context $\mathbf{c}_i$, we introduce a lightweight router that projects it onto K latent context prototypes, producing a routing distribution:

$$\mathbf{r}_i = \mathrm{softmax}(\phi_r(\mathbf{c}_i)), \qquad \xi_i = -\frac{1}{\log K}\sum_{k=1}^{K} r_{ik}\log(r_{ik}+\epsilon). \tag{8}$$

Here, ϕ_r is the routing projection, $\mathbf{r}_i$ is the routing distribution over the K prototypes, and ϵ is a small constant for numerical stability. The normalized entropy ξ_i serves as an uncertainty score. Intuitively, when $\mathbf{c}_i$ aligns well with a specific prototype, the routing distribution is sharp (low entropy), indicating a confident and self-consistent context. In contrast, a diffuse distribution (high entropy) suggests that the context does not clearly match any known pattern, indicating potential distribution shift or insufficient information.

Uncertainty-Adaptive Neighbor Support. Instead of applying uniform graph propagation, REACT adaptively controls spatial interaction based on estimated uncertainty. Let $\tilde{\mathbf{A}}$ denote the row-normalized support matrix. We first compute a neighborhood-supported context $\bar{\mathbf{c}}_i = \sum_{j=1}^{N} \tilde{A}_{ij}\mathbf{c}_j$.

The uncertainty score is then converted into a node-specific support strength,

$$\alpha_i = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\xi_i, \tag{9}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are the lower and upper bounds of the support strength. The calibrated context is given by

$$\mathbf{c}_i^{\star} = \mathrm{LN}\big((1-\alpha_i)\mathbf{c}_i + \alpha_i\bar{\mathbf{c}}_i\big). \tag{10}$$

This design addresses uncontrolled propagation. When local context is reliable (low ξ_i), the model relies on node-local information, avoiding contamination from potentially shifted neighbors. When local context is uncertain (high ξ_i), the model increases neighborhood support to stabilize estimation.

Importantly, spatial interaction is no longer a default operation but a reliability-dependent correction mechanism. **Appendix B.3** further analyzes this adaptive mixture and bounds the calibrated context error in terms of local estimation error, propagated neighbor error, and neighborhood mismatch.

4.4 Gated Context Modulation and Prediction

Gated Context Modulation. The calibrated context should refine the predictive anchor without disrupting its graph-free semantics. To this end, REACT applies *gated feature-wise modulation* rather than introducing a separate context-driven predictor [25]. This design ensures that context serves as a controlled adjustment signal instead of an alternative prediction pathway.

Given the calibrated context $\mathbf{c}_i^{\star}$, feature-wise affine parameters are generated as

$$(\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i) = \phi_a(\mathbf{c}_i^{\star}), \tag{11}$$

and a modulation gate is computed jointly from the predictive anchor and the calibrated context,

$$\mathbf{g}_i = \mathrm{sigmoid}\Big(\phi_g\big([\mathrm{LN}(\mathbf{h}_i),\ \mathbf{c}_i^{\star}]\big)\Big). \tag{12}$$

The modulated hidden state is then

$$\tilde{\mathbf{h}}_i = \mathrm{LN}\Big(\mathbf{h}_i + \mathbf{g}_i \odot \big(\mathrm{LN}(\mathbf{h}_i)\odot\tanh(\boldsymbol{\gamma}_i) + \boldsymbol{\beta}_i\big)\Big). \tag{13}$$

This mechanism explicitly controls how context influences prediction. The affine parameters determine *how* each feature should be adjusted under the current context, while the gate $\mathbf{g}_i$ determines *how much* adjustment should be applied. When context is reliable and informative, stronger modulation is activated; when context is less trustworthy, the gate suppresses its influence, preserving the stability of the predictive anchor.

As a result, context does not overwrite node-local prediction but acts as a bounded and reliability-aware correction signal. **Appendix B.4** further shows that, under mild Lipschitz conditions, the prediction perturbation induced by context error is bounded by the calibrated context error.

Final Prediction. The final multi-step forecast is obtained by decoding the modulated hidden state,

$$\hat{\mathbf{y}}_i = \phi_o(\tilde{\mathbf{h}}_i) \in \mathbb{R}^H. \tag{14}$$

Here, ϕ_o is the prediction head that maps the modulated hidden state to the H-step forecast. This design completes the separation between prediction and adaptation: the graph-free encoder determines *what to predict* based on node-local dynamics, while the calibrated context determines *how to adjust* the prediction under distribution shift.

4.5 Training Objective

We train REACT with masked MAE on valid horizon-node pairs. In addition, we penalize excessive reliance on neighbor support by regularizing the support strength,

$$\mathcal{L}_{\text{pred}} = \frac{1}{|\Omega|} \sum_{(h,i)\in\Omega} |\hat{Y}_{h,i} - Y_{h,i}|, \qquad \mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{ctx}} \cdot \frac{1}{N} \sum_{i=1}^{N} \alpha_i^2. \tag{15}$$

The regularizer does not force the model to ignore neighborhood information. Instead, it encourages REACT to use structural support only when the local context is sufficiently uncertain to justify it.

To improve robustness to missing auxiliary information, REACT adopts prior masking during training. Let ρ be the masking ratio and $\mathbf{m}$ the node-wise Bernoulli mask. The training objective is

$$\min_{\theta} \mathbb{E}_{\mathbf{m}\sim\text{Bernoulli}(1-\rho)} \Big[\mathcal{L}\big(f_\theta(\mathbf{X}, \mathbf{m} \odot \mathbf{P}, \mathbf{m}), \mathbf{Y}\big) \Big]. \tag{16}$$

This strategy improves robustness to incomplete or unreliable auxiliary information by explicitly exposing the model to missing-prior scenarios during training. The overall model training **complexity analysis** is shown in **Appendix B.5**.

5 Experiments

We structure the evaluation around six research questions (RQs). RQ1–RQ5 are reported in the main text, while RQ6 is provided in **Appendix D.4**.

RQ1. *How does REACT perform compared with state-of-the-art baselines under OOD shifts?*
RQ2. *How robust is REACT under joint and isolated spatio-temporal OOD shifts?*
RQ3. *Which core components most contribute to the generalization improvements of REACT?*
RQ4. *Does the learned reliability signal align with prediction error and neighbor support?*
RQ5. *Does REACT achieve superior computational efficiency compared with existing OOD baselines?*
RQ6. *How does the context dimension affect the forecasting performance of REACT?*

5.1 Experimental Settings

Datasets and OOD protocol. We evaluate REACT on ten real datasets under a unified spatio-temporal OOD protocol. It includes 4 LargeST traffic networks [21] (SD, GBA, GLA, and CA) and 6 smaller urban datasets [32] covering pedestrian, taxi, bike, traffic, and service-request signals. The protocol trains models on an initial temporal period and a retained training graph, and evaluates them on subsequent periods with some training nodes removed and unseen nodes added. **Dataset statistics, split configurations, and distribution shift visualizations** are provided in **Appendix C.1**.

Baselines. We compare REACT against two categories of baselines: independent and identically distributed (IID) forecasting models and OOD-specific methods. **(i) The IID group** includes representative graph convolutional, dynamic graph, identity-aware, scalable, and mixer-based architectures,

Table 1: Performance comparison under OOD settings. MAPE is reported in percentage. '–' indicates out-of-memory. The best and second-best results are marked as **bold** and underlined.

Method	SD			GBA			GLA			CA			311NYC		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	27.60	42.60	22.47	28.24	43.57	23.79	29.63	45.01	21.75	26.81	41.95	26.29	2.36	3.52	67.65
GWNet	32.08	48.64	25.64	31.17	46.03	25.85	32.71	48.68	23.43	29.79	45.65	28.94	2.33	3.50	66.58
D2STGNN	28.19	42.76	23.27	27.51	41.65	23.34	–			–			2.33	3.47	69.95
STID	26.83	41.87	20.18	27.67	43.04	24.68	28.89	44.59	20.31	27.34	43.07	33.52	2.22	3.35	64.83
BigST	26.79	41.46	19.96	30.07	45.44	24.50	29.47	45.09	20.55	27.42	42.77	22.09	2.21	3.33	65.29
RPMixer	31.65	51.18	22.33	33.44	52.17	26.85	34.39	55.29	24.22	32.49	53.21	25.74	2.35	3.54	65.40
CaST	30.74	45.91	31.18	32.73	48.05	35.29	37.57	54.62	37.41	33.28	49.10	48.73	2.37	3.36	81.12
CauSTG	28.87	43.84	23.29	28.74	43.07	24.37	29.53	44.46	21.06	27.67	42.77	27.82	2.24	3.36	67.47
STONE	27.61	42.81	23.70	29.50	44.73	29.46	–			–			2.36	3.59	62.74
STEVE	28.09	43.15	22.90	28.29	43.38	24.70	29.27	44.69	25.22	27.92	46.36	41.40	2.48	3.70	71.24
STOP	25.99	40.65	21.86	26.12	40.55	23.72	29.44	44.92	25.51	26.16	40.10	28.13	2.19	3.33	66.58
REACT	**23.43**	**37.58**	**18.84**	**25.31**	**39.25**	**20.15**	**25.75**	**39.91**	**18.17**	**24.03**	**37.61**	**20.68**	**2.16**	**3.29**	**61.69**

Method	BikeCHI			FlowSC			PedZurich			SpeedNYC			TaxiCHI		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	1.36	2.74	50.14	17.72	28.23	11.68	40.90	71.12	81.58	5.41	8.58	29.22	16.09	55.35	58.48
GWNet	1.48	3.16	51.46	18.88	29.50	13.86	41.89	73.67	87.36	5.51	8.54	29.74	15.12	49.60	55.58
D2STGNN	1.42	2.72	56.41	16.70	26.77	11.42	39.88	70.54	80.48	5.60	8.66	29.33	15.81	51.17	56.28
STID	1.47	2.88	56.03	18.22	28.78	11.57	44.72	80.24	113.66	5.80	8.91	31.18	26.06	91.69	75.98
BigST	1.39	2.87	50.79	16.58	27.03	10.95	37.00	66.33	80.35	5.59	8.67	29.77	15.50	48.20	57.80
RPMixer	1.56	2.97	60.26	18.21	29.19	11.60	44.75	75.52	151.05	5.85	9.24	29.08	16.74	48.93	86.23
CaST	1.39	2.96	49.46	25.28	35.08	33.13	52.73	94.35	121.19	9.61	12.31	68.58	18.17	56.90	53.70
CauSTG	1.50	2.96	57.57	20.87	31.38	15.91	46.76	80.50	137.50	6.17	9.56	33.92	27.50	97.63	94.11
STONE	1.50	2.89	59.11	22.38	32.87	13.94	54.49	91.39	190.00	6.03	9.02	35.51	31.33	111.41	79.64
STEVE	1.41	2.88	54.05	19.30	29.90	13.46	44.95	77.96	149.62	5.53	8.54	29.68	16.42	53.20	54.24
STOP	1.32	2.71	49.10	18.67	28.93	13.96	36.92	63.93	105.20	5.37	8.46	28.37	14.80	47.88	50.95
REACT	**1.28**	**2.44**	**48.97**	**15.49**	**25.71**	**9.68**	**32.96**	**58.41**	**79.82**	**5.20**	**8.17**	**27.44**	**14.62**	**47.28**	**50.26**

such as STGCN [38], GWNet [35], D2STGNN [28], STID [27], BigST [13], and RPMixer [37]. **(ii) The OOD-specific group** includes CaST [36], CauSTG [45], STONE [31], STEVE [16], and STOP [23]. All methods are trained and selected using their respective validation splits under the same OOD protocol. **Baseline details** and **implementation** are provided in **Appendix C.2** and **Appendix C.3**.

Protocols and metrics. We report predictive accuracy using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), where lower values indicate better performance. Computational efficiency is evaluated in terms of Latency, Epoch time, FLOPs, and Memory usage. For node-level evaluation, All denotes all test nodes, while UnF denotes unseen future nodes that appear at test time. The main text presents aggregated results, while detailed year-wise and protocol-specific results are provided in the Appendix. **Formal metric definitions** are included in **Appendix C.4**.

5.2 Overall Performance Comparison (*RQ1*)

The main OOD forecasting results are reported in Table 1. First, on the four large-scale datasets, REACT reduces MAE, RMSE, and MAPE over the strongest baseline by 8.00%, 6.80%, and 9.05% on average, respectively, showing stable gains even on the largest CA dataset. Second, existing OOD-specific baselines show uneven robustness. STOP performs competitively on several datasets, while STID remains competitive on GLA, and several OOD methods fall behind strong IID models on small-scale datasets. This suggests that invariant learning, causal adjustment, or environment augmentation can reduce part of the shift but may still trust biased local states when node context changes. In contrast, REACT explicitly estimates context uncertainty and uses reliability-guided neighbor support to correct uncertain node states. Third, REACT achieves the best MAE on all six small-scale datasets and reduces MAE over the strongest baseline by 4.35% on average. This indicates that the same calibration mechanism transfers from large traffic networks to diverse urban signals. Complete results can be found in **Appendix D.1**.

5.3 Robustness under OOD Shifts (*RQ2*)

We further evaluate robustness under stronger spatial distribution shifts and isolated OOD settings.

Fig. 3 analyzes the joint OOD setting by varying the spatial shift ratio on GLA and CA and reporting UnF MAE. This setting directly stresses unseen-node forecasting, as increasing spatial shift ratios amplify the mismatch between training and test graphs. Across both datasets, REACT consistently achieves the lowest UnF error under all shift ratios, while competing OOD baselines exhibit larger errors and less stable trends as the shift intensity increases. These results suggest that REACT's reliability-guided context calibration remains effective under stronger unseen-node distribution shifts, rather than merely improving average performance.

Fig. 4 evaluates isolated spatial-only OOD (SOOD) and temporal-only OOD (TOOD) settings on GLA, using UnF MAE for SOOD and MAE for TOOD. In SOOD, REACT remains the best performer as r_s increases from 0.1 to 0.3, with UnF MAE varying only slightly from 23.61 to 23.99, while STOP and CauSTG remain consistently worse. In TOOD, REACT also achieves the best performance across target years, with MAE ranging from 23.28 to 25.68. These results indicate that reliability-guided context support effectively handles both structural and temporal distribution shifts. Full results are provided in **Appendix D.2**.

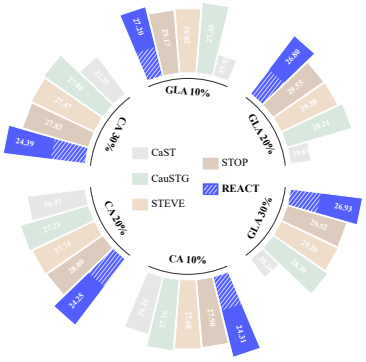

Figure 3: Performance on unseen nodes under varying degrees of spatial shift.

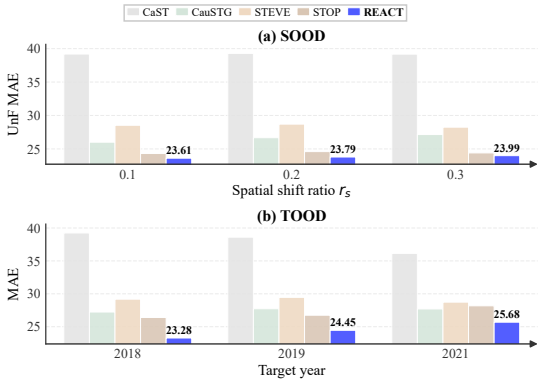

Figure 4: Robustness under isolated spatial and temporal OOD shifts.

5.4 Ablation Study (*RQ3*)

We conduct ablation studies to assess the contribution of five key components in REACT: (1) **w/o Context**, which removes the entire context modeling branch; (2) **w/o Dynamic**, which removes observation-driven context statistics; (3) **w/o Static**, which removes static prior information; (4) **w/o Support**, which disables neighbor-based context calibration; and (5) **w/o Gate**, which removes the node-specific modulation gate and applies context modulation without adaptive amplitude control.

Fig. 5 reports results on representative datasets. The w/o Context variant exhibits the largest degradation, confirming that explicit context estimation is the main source of robustness under joint spatio-temporal shifts. Removing static priors also causes a clear accuracy drop, while removing dynamic statistics leads to smaller but consistent degradation, indicating that prior knowledge and observation-driven cues provide complementary context evidence. The w/o Support variant degrades both All and UnF performance, with a larger drop on GBA than on GLA, showing that unreliable local context benefits from reliability-guided neighbor calibration. Finally, w/o Gate consistently underperforms REACT on both All and UnF nodes, demonstrating that the calibrated context should modulate the prediction anchor through a node-specific gate rather than being applied with uncontrolled strength. Overall, these results validate the full context-estimation, reliability-calibration, and gated-modulation design.

5.5 Reliability Diagnostics (*RQ4*)

Fig. 6 examines whether the learned reliability signal aligns with model behavior beyond ablation improvements. Higher routing uncertainty is consistently associated with larger prediction errors: binned MAE rises monotonically from 13.84 to 32.06, with Pearson and Spearman correlations of 0.59 and 0.66, respectively. Since α_i is derived from node-specific uncertainty, we further group test nodes by learned support strength and compare REACT with its no-support variant within each

group. The resulting MAE changes are small and non-monotonic across bins, indicating that stronger support alone does not guarantee larger gains. Instead, context calibration should be interpreted as a risk-controlled trade-off between local uncertainty and neighborhood mismatch, consistent with the reliability-guided support analysis in **Appendix B.3**.

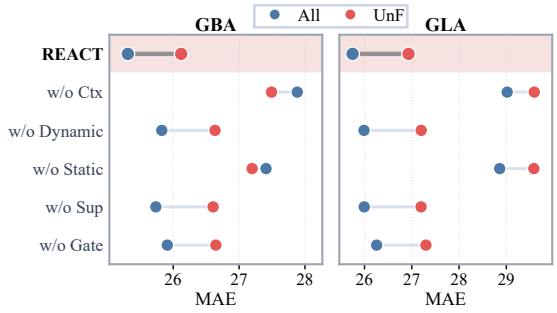

Figure 5: Ablation study on GBA and GLA datasets.

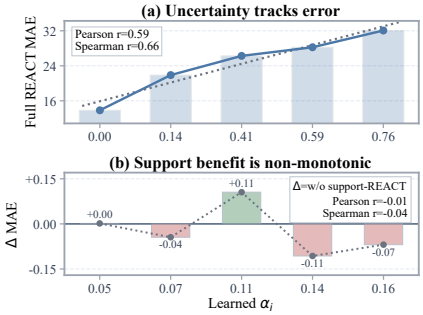

Figure 6: Reliability diagnostics.

5.6 Efficiency Study (*RQ5*)

Table 2 reports MAE together with efficiency performance in terms of Latency, Epoch, FLOPs, and Memory on representative datasets. Among OOD baselines, REACT attains the lowest value in all 15 reported entries. Compared with the strongest overall OOD baseline STOP, REACT reduces Latency, Epoch, FLOPs, and Memory by 56.80%, 52.73%, 86.54%, and 75.57% on average, respectively. These savings indicate that REACT's robustness does not rely on expensive graph computation; instead, its graph-free prediction anchor and lightweight context-calibration path yield a better accuracy–efficiency trade-off for large-scale OOD forecasting.

Table 2: Efficiency comparison. '–' indicates out-of-memory. Best results are marked as **bold**.

	SD					GBA					GLA				
Method	**MAE**	**Latency**	**Epoch**	**FLOPs**	**Memory**	**MAE**	**Latency**	**Epoch**	**FLOPs**	**Memory**	**MAE**	**Latency**	**Epoch**	**FLOPs**	**Memory**
CaST	30.74	13 ms	54 s	15 G	1652 MB	32.73	36 ms	139 s	50 G	5457 MB	37.57	59 ms	214 s	81 G	8884 MB
CauSTG	28.87	43 ms	109 s	854 G	2302 MB	28.74	232 ms	601 s	8398 G	7849 MB	29.53	489 ms	502 s	21940 G	13208 MB
STONE	27.61	101 ms	142 s	277 G	9046 MB	29.50	182 ms	544 s	619 G	15427 MB	–	–	–	–	–
STEVE	28.09	9 ms	37 s	88 G	1107 MB	28.29	32 ms	90 s	719 G	3691 MB	29.27	60 ms	179 s	1807 G	6100 MB
STOP	25.99	7 ms	47 s	87 G	1698 MB	26.12	22 ms	119 s	286 G	5560 MB	29.44	32 ms	184 s	467 G	9062 MB
REACT	**23.43**	**4 ms**	**24 s**	**8 G**	**415 MB**	**25.31**	**7 ms**	**53 s**	**42 G**	**1281 MB**	**25.75**	**13 ms**	**85 s**	**77 G**	**2339 MB**

6 Conclusion

This paper studies spatio-temporal OOD forecasting and proposes REACT, a lightweight reliability-aware context-calibrated framework. By decoupling prediction from context adaptation, REACT establishes a graph-free predictive anchor and uses graph interaction as a selective, reliability-guided correction mechanism. Experiments on ten real-world datasets demonstrate that REACT consistently outperforms representative baselines in both accuracy and efficiency. These results highlight a key insight: under OOD shifts, improving robustness is not only about learning invariant representations, but also about explicitly modeling and controlling the reliability of context.

Acknowledgments and Disclosure of Funding

This work was supported in part by the NSFC under Grants No. (62402422 and U23A20296), Ningbo NSF under Grant No. 2025J007, Zhejiang Provincial NSFC under Grant No. LZ25F020001, the National Science and Technology Major Project No. 2026ZD0128300, Zhejiang Province's "Jianbing" R&D Project under Grants No. 2025C01195, and Yongjiang Talent Introduction Programme (2024A-162-G). Ziquan Fang is the corresponding author. The authors declare no competing interests.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant Risk Minimization. *arXiv preprint arXiv:1907.02893* (2019).

[2] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *Advances in Neural Information Processing Systems*, Vol. 33. 17804–17815.

[3] Wei Chen and Yuxuan Liang. 2025. Expand and Compress: Exploring Tuning Principles for Continual Spatio-Temporal Graph Forecasting. In *International Conference on Learning Representations*.

[4] Xu Chen, Junshan Wang, and Kunqing Xie. 2021. TrafficStream: A Streaming Traffic Flow Forecasting Framework Based on Graph Neural Networks and Continual Learning. arXiv:2106.06273

[5] Junyoung Chung, Caglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] Jerome T. Connor, Robert D. Martin, and Les E. Atlas. 1994. Recurrent Neural Networks and Robust Time Series Prediction. *IEEE Transactions on Neural Networks* 5, 2 (1994), 240–254.

[7] Noel Cressie and Christopher K. Wikle. 2011. *Statistics for Spatio-Temporal Data*. Wiley.

[8] Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. 2024. Heterogeneity-Informed Meta-Parameter Learning for Spatiotemporal Time Series Forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 631–641.

[9] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1996. Support Vector Regression Machines. In *Advances in Neural Information Processing Systems 9*, Vol. 9. 155–161.

[10] Yifan Duan, Jian Zhao, Pengcheng, Junyuan Mao, Hao Wu, Jingyu Xu, Shilong Wang, Caoyuan Ma, Kai Wang, Kun Wang, and Xuelong Li. 2024. Causal Deciphering and Inpainting in Spatio-Temporal Dynamics via Diffusion Model. In *Advances in Neural Information Processing Systems*, Vol. 37. 107604–107632.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* 17, 59 (2016), 1–35.

[12] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 922–929.

[13] Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. 2024. BigST: Linear Complexity Spatio-Temporal Graph Neural Network for Traffic Forecasting on Large-Scale Road Networks. *Proceedings of the VLDB Endowment* 17, 5 (2024), 1081–1090.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[15] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-Temporal Self-Supervised Learning for Traffic Flow Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4356–4364.

[16] Jiahao Ji, Wentao Zhang, Jingyuan Wang, and Chao Huang. 2025. Seeing the Unseen: Learning Basis Confounder Representations for Robust Traffic Prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 577–588.

[17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[18] Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. 2023. GPT-ST: Generative Pre-Training of Spatio-Temporal Graph Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 36. 70229–70246.

[19] Aoyu Liu and Yaying Zhang. 2026. A General Spatio-Temporal Backbone with Scalable Contextual Pattern Bank for Urban Continual Forecasting. In *International Conference on Learning Representations*.

[20] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. 2023. 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*. 4125–4129.

[21] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2023. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. In *Advances in Neural Information Processing Systems*, Vol. 36.

[22] Helmut Lütkepohl. 2005. *New Introduction to Multiple Time Series Analysis*. Springer.

[23] Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. 2025. Robust Spatio-Temporal Centralized Interaction for OOD Learning. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Vol. 267. 42165–42192.

[24] Minbo Ma, Kai Tang, Huan Li, Fei Teng, Dalin Zhang, and Tianrui Li. 2025. Beyond Fixed Variables: Expanding-variate Time Series Forecasting via Flat Scheme and Spatio-temporal Focal Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. 2054–2065.

[25] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[26] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. In *International Conference on Learning Representations*.

[27] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 4454–4458.

[28] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S. Jensen. 2022. Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2733–2746.

[29] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, Vol. 28.

[30] Yongfeng Su, Hongwen Li, Zijian Zhang, Ziquan Fang, Lu Chen, Christian S. Jensen, Hong Gao, and Yinjun Han. 2026. SqLinear: Balanced Square Partitioning Makes Linear Interaction Sufficient for Large-Scale Traffic Forecasting. *arXiv preprint arXiv:2606.21072* (2026).

[31] Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. 2024. STONE: A Spatio-temporal OOD Learning Framework Kills Both Spatial and Temporal Shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2948–2959.

[32] Hongjun Wang, Jiyuan Chen, Tong Pan, Zheng Dong, Renhe Jiang, and Xuan Song. 2025. Evaluating the Generalization Ability of Spatiotemporal Model in Urban Scenario. *IEEE Transactions on Mobile Computing* 24, 12 (2025), 13535–13548.

[33] Kun Wang, Hao Wu, Yifan Duan, Guibin Zhang, Kai Wang, Xiaojiang Peng, Yu Zheng, Yuxuan Liang, and Yang Wang. 2024. NuwaDynamics: Discovering and Updating in Causal Spatio-Temporal Modeling. In *The Twelfth International Conference on Learning Representations*.

[34] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 753–763.

[35] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 1907–1913.

[36] Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. 2023. Deciphering Spatio-Temporal Graph Forecasting: A Causal Lens and Treatment. In *Advances in Neural Information Processing Systems*, Vol. 36. 37068–37088.

[37] Chin-Chia Michael Yeh, Yujie Fan, Xin Dai, Uday Singh Saini, Vivian Lai, Prince Osei Aboagye, Junpeng Wang, Huiyuan Chen, Yan Zheng, Zhongfang Zhuang, Liang Wang, and Wei Zhang. 2024. RPMixer: Shaking Up Time Series Forecasting with Random Projections for Large Spatial-Temporal Data. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3919–3930.

[38] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 3634–3640.

[39] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4095–4106.

[40] Haoyu Zhang, Wentao Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. 2025. STRAP: Spatio-Temporal Pattern Retrieval for Out-of-Distribution Generalization. In *Advances in Neural Information Processing Systems*, Vol. 38.

[41] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[42] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-Based Prediction Model for Spatio-Temporal Data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 92:1–92:4.

[43] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1234–1241.

[44] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2023. Domain Generalization: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2023), 4396–4415.

[45] Zhengyang Zhou, Qihe Huang, Kuo Yang, Kun Wang, Xu Wang, Yudong Zhang, Yuxuan Liang, and Yang Wang. 2023. Maintaining the Status Quo: Capturing Invariant Relations for OOD Spatiotemporal Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3603–3614.

SUPPLEMENTARY MATERIAL

REACT: A LIGHTWEIGHT RELIABILITY-AWARE FRAMEWORK FOR SPATIO-TEMPORAL OUT-OF-DISTRIBUTION PREDICTION

TABLE OF CONTENTS

A Notations and Descriptions **15**

B Theoretical Analysis **15**

B.1 Temporal OOD Induces Context Misalignment 15

B.2 Spatial OOD Causes Information Loss in Context Inference 16

B.3 Reliability-Guided Calibration Balances Local and Neighbor Errors 17

B.4 Gated Context Modulation Ensures Stable Prediction Updates 18

B.5 Computational Complexity 19

C Experimental Setup **19**

C.1 Evaluation Protocols and Datasets 19

C.2 Baseline Families and OOD Adaptation 20

C.3 Implementation Details 21

C.4 Metric Definitions 21

D Extended Experimental Results **21**

D.1 Complete Results 21

D.2 Isolated-Shift Robustness 22

D.3 Static Metadata and Baseline Fairness 23

D.4 Hyperparameter Sensitivity (*RQ6*) 24

E More Discussion **25**

E.1 Relation to Retrieval-Augmented and Continual Paradigms 25

E.2 Limitations 26

E.3 Future Work 26

E.4 Broader Impacts 26

A Notations and Descriptions

Table 3 presents the frequently used notations and their descriptions.

Table 3: Notations and descriptions

Notation	Description
$\mathcal{V}^e, N^e$	Set of nodes and number of nodes in environment e
$\mathbf{A}^e, \tilde{\mathbf{A}}$	Adjacency matrix and row-normalized support matrix
$\mathcal{G}^e$	Spatial graph $\mathcal{G}^e = (\mathcal{V}^e, \mathbf{A}^e)$
$\mathbf{X}$	Historical spatio-temporal observations
$\mathbf{Y}, \hat{\mathbf{Y}}$	Future and predicted spatio-temporal sequences
T, H, d	Input length, prediction horizon, and feature dimension
$\mathbf{s}_i, \boldsymbol{\omega}_i$	Target history and context descriptor of node i
$\mathbf{p}_i, m_i, \rho$	Node prior, prior indicator, and masking ratio
$\mathbf{h}_i, \tilde{\mathbf{h}}_i$	Predictive anchor and modulated hidden state
$\mathbf{c}_i, \bar{\mathbf{c}}_i, \mathbf{c}_i^\star$	Raw, neighbor-supported, and calibrated contexts
ξ_i, α_i	Routing uncertainty and support strength
$\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i, \mathbf{g}_i$	Modulation parameters and gate

B Theoretical Analysis

This section provides a theoretical justification for the central design of REACT: context reliability is fragile under OOD shifts, and must be explicitly estimated and controlled.

Rather than assuming that OOD shifts always destroy context, we identify two fundamental failure modes: (i) temporal shifts that alter the environment-conditional meaning of context, and (ii) spatial shifts that degrade the observable evidence for context inference. We then show how REACT mitigates these risks through reliability-guided calibration and controlled context modulation.

B.1 Temporal OOD Induces Context Misalignment

Let $\mathbf{c}_i^\dagger$ denote the ideal predictive context of node i, and let $\mathbf{u}_i = [\boldsymbol{\omega}_i, \mathbf{p}_i, m_i]$ denote the observable evidence used to infer this context, where $\boldsymbol{\omega}_i$ is the observation descriptor in Eq. (6). For an environment e, define the squared context risk of an estimator g as

$$R_e(g) = \mathbb{E}_e \left[\left\| g(\mathbf{u}_i) - \mathbf{c}_i^\dagger \right\|_2^2 \right]. \tag{17}$$

The Bayes-optimal context estimator under this risk is

$$g_e^\star(\mathbf{u}) = \mathbb{E} \left[\mathbf{c}_i^\dagger \mid \mathbf{u}_i = \mathbf{u}, e \right]. \tag{18}$$

We evaluate the deployed source estimator on target evidence values. When the source and target evidence supports do not perfectly overlap, $g_s^\star$ should be understood as the source-trained estimator extended to the target evidence domain, since this is the function actually used at deployment.

Proposition 1. Let s be the source environment and t be the target environment. Under squared context risk,

$$R_t(g_s^\star) - R_t(g_t^\star) = \mathbb{E}_t \left[\| g_s^\star(\mathbf{u}_i) - g_t^\star(\mathbf{u}_i) \|_2^2 \right]. \tag{19}$$

Therefore, if $g_s^\star(\mathbf{u}) \neq g_t^\star(\mathbf{u})$ on a set with positive target probability, then using the source-optimal context estimator incurs strictly positive excess target context risk.

Proof. For compactness, write $\mathbf{c} = \mathbf{c}_i^\dagger$ and $\mathbf{u} = \mathbf{u}_i$. Expanding the target risk of $g_s^\star$ around the target Bayes estimator gives

$$\begin{aligned} R_t(g_s^\star) &= \mathbb{E}_t \left[\| g_s^\star(\mathbf{u}) - \mathbf{c} \|_2^2 \right] \\ &= \mathbb{E}_t \left[\| g_s^\star(\mathbf{u}) - g_t^\star(\mathbf{u}) \|_2^2 \right] + R_t(g_t^\star) \\ &\quad + 2\mathbb{E}_t \left[\langle g_s^\star(\mathbf{u}) - g_t^\star(\mathbf{u}), g_t^\star(\mathbf{u}) - \mathbf{c} \rangle \right]. \end{aligned} \tag{20}$$

The cross term is zero because

$$\mathbb{E}_t\left[g_t^\star(\mathbf{u}) - \mathbf{c} \mid \mathbf{u}\right] = g_t^\star(\mathbf{u}) - \mathbb{E}_t[\mathbf{c} \mid \mathbf{u}] = \mathbf{0}. \tag{21}$$

Subtracting $R_t(g_t^\star)$ yields Eq. (19). The strict positivity condition follows because the integrand is nonnegative and is positive on a set with positive target probability.

Proposition 1 shows that temporal OOD causes context unreliability when the conditional mean $\mathbb{E}[\mathbf{c}_i^\dagger \mid \mathbf{u}_i, e]$ changes across environments. In this case, the same observable evidence supports different target-optimal context estimates in the source and target regimes. The resulting environment-induced context bias can be written as

$$\mathbf{b}_t(\mathbf{u}) = g_s^\star(\mathbf{u}) - g_t^\star(\mathbf{u}), \tag{22}$$

and the excess risk equals $\mathbb{E}_t\|\mathbf{b}_t(\mathbf{u}_i)\|_2^2$.

B.2 Spatial OOD Causes Information Loss in Context Inference

Spatial OOD can also make context unreliable by degrading the evidence available for context inference. Let $\mathbf{u}_i$ be the full node evidence and let $\tilde{\mathbf{u}}_i$ be a degraded observation generated from an environment-dependent degradation channel $S_e(\tilde{\mathbf{u}} \mid \mathbf{u})$. This covers missing priors, masked priors, noisy priors, and semantically shifted priors. The prior-masking mechanism used by REACT is one concrete case:

$$\tilde{\mathbf{u}}_i = [\boldsymbol{\omega}_i, m_i\mathbf{p}_i, m_i]. \tag{23}$$

We assume that the degraded observation does not contain additional information beyond the full evidence, i.e., $\mathbf{c}_i^\dagger \perp \tilde{\mathbf{u}}_i \mid \mathbf{u}_i, e$.

Proposition 2. Define the optimal full-evidence and degraded-evidence risks as

$$R_e^{\text{full}} = \inf_g \mathbb{E}_e\left[\left\|g(\mathbf{u}_i) - \mathbf{c}_i^\dagger\right\|_2^2\right], \tag{24}$$

$$R_e^{\text{deg}} = \inf_{\tilde{g}} \mathbb{E}_e\left[\left\|\tilde{g}(\tilde{\mathbf{u}}_i) - \mathbf{c}_i^\dagger\right\|_2^2\right]. \tag{25}$$

Then

$$R_e^{\text{deg}} - R_e^{\text{full}} = \mathbb{E}_e\left[\operatorname{tr}\left(\operatorname{Cov}\left(\mathbb{E}[\mathbf{c}_i^\dagger \mid \mathbf{u}_i, e] \mid \tilde{\mathbf{u}}_i, e\right)\right)\right] \geq 0. \tag{26}$$

The inequality is strict whenever the full evidence contains predictive context information that is not recoverable from the degraded observation.

Proof. For a random vector $\mathbf{c}$, the minimum mean-squared estimation risk given an observation is the expected trace of the conditional covariance. Therefore,

$$R_e^{\text{full}} = \mathbb{E}_e\left[\operatorname{tr}\left(\operatorname{Cov}(\mathbf{c}_i^\dagger \mid \mathbf{u}_i, e)\right)\right], \tag{27}$$

$$R_e^{\text{deg}} = \mathbb{E}_e\left[\operatorname{tr}\left(\operatorname{Cov}(\mathbf{c}_i^\dagger \mid \tilde{\mathbf{u}}_i, e)\right)\right]. \tag{28}$$

By the law of total covariance conditioned on $\tilde{\mathbf{u}}_i$, and using $\mathbf{c}_i^\dagger \perp \tilde{\mathbf{u}}_i \mid \mathbf{u}_i, e$, we have

$$\begin{aligned}\operatorname{Cov}(\mathbf{c}_i^\dagger \mid \tilde{\mathbf{u}}_i, e) = \mathbb{E}\left[\operatorname{Cov}(\mathbf{c}_i^\dagger \mid \mathbf{u}_i, e) \mid \tilde{\mathbf{u}}_i, e\right] \\ + \operatorname{Cov}\left(\mathbb{E}[\mathbf{c}_i^\dagger \mid \mathbf{u}_i, e] \mid \tilde{\mathbf{u}}_i, e\right).\end{aligned} \tag{29}$$

Taking the trace and then expectation over $\tilde{\mathbf{u}}_i$ gives Eq. (26). Since covariance matrices are positive semidefinite, the trace term is nonnegative. It is strictly positive when $\mathbb{E}[\mathbf{c}_i^\dagger \mid \mathbf{u}_i, e]$ still varies after conditioning on $\tilde{\mathbf{u}}_i$, which means that the degraded observation has lost context-relevant information.

Corollary 1 (Limit of unconditional context trust). Consider a model that relies on a source-trained context estimator at deployment without assessing target-side context reliability. If either (i) the target-optimal context estimator differs from the source-optimal estimator on a set with positive target probability, or (ii) deployment provides degraded evidence that loses context-relevant information, then unconditional context trust incurs positive excess or irreducible context risk.

Proof. Case (i) follows directly from Proposition 1, and case (ii) follows directly from Proposition 2.

B.3 Reliability-Guided Calibration Balances Local and Neighbor Errors

The previous two propositions explain why the raw local context estimate may be unreliable under OOD shift. REACT addresses this risk by injecting neighborhood support only through a reliability-guided mixing weight. To keep notation aligned with Section 4, we distinguish the raw context $\mathbf{c}_i$ in Eq. (7), the neighbor-supported context $\bar{\mathbf{c}}_i = \sum_j \tilde{A}_{ij}\mathbf{c}_j$, and the calibrated context $\mathbf{c}_i^\star$ in Eq. (10). The following result analyzes the pre-normalization mixture

$$\tilde{\mathbf{c}}_i := (1-\alpha_i)\mathbf{c}_i + \alpha_i \bar{\mathbf{c}}_i, \tag{30}$$

while the subsequent layer normalization is included in the downstream Lipschitz maps defined in Section B.4.

For node i, let $\mathbf{c}_i^\dagger \in \mathbb{R}^{d_c}$ denote the ideal latent context sufficient for prediction, and let the raw context estimation error satisfy

$$\mathbf{c}_i = \mathbf{c}_i^\dagger + \boldsymbol{\varepsilon}_i. \tag{31}$$

Let $\tilde{A} \in \mathbb{R}^{N\times N}$ be the row-normalized adjacency matrix used in Eq. (10), where $\tilde{A}_{ij} \geq 0$ and $\sum_j \tilde{A}_{ij} = 1$. REACT forms $\bar{\mathbf{c}}_i = \sum_j \tilde{A}_{ij}\mathbf{c}_j$ and

$$\tilde{\mathbf{c}}_i = (1-\alpha_i)\mathbf{c}_i + \alpha_i \sum_j \tilde{A}_{ij}\mathbf{c}_j, \tag{32}$$

where $\alpha_i \in [0,1]$ is the entropy-aware support strength defined in Eq. (9).

Proposition 3 (Context Calibration Error Bound). Define the neighborhood mismatch term

$$\eta_i := \left\| \sum_j \tilde{A}_{ij}\mathbf{c}_j^\dagger - \mathbf{c}_i^\dagger \right\|_2. \tag{33}$$

Then the pre-normalization calibrated context error is bounded by

$$\|\tilde{\mathbf{c}}_i - \mathbf{c}_i^\dagger\|_2 \leq (1-\alpha_i)\|\boldsymbol{\varepsilon}_i\|_2 + \alpha_i \sum_j \tilde{A}_{ij}\|\boldsymbol{\varepsilon}_j\|_2 + \alpha_i\eta_i. \tag{34}$$

Proof. Substituting $\mathbf{c}_j = \mathbf{c}_j^\dagger + \boldsymbol{\varepsilon}_j$ into the definition of $\tilde{\mathbf{c}}_i$ gives

$$\begin{aligned}\tilde{\mathbf{c}}_i - \mathbf{c}_i^\dagger &= (1-\alpha_i)(\mathbf{c}_i - \mathbf{c}_i^\dagger) + \alpha_i \sum_j \tilde{A}_{ij}(\mathbf{c}_j - \mathbf{c}_i^\dagger) \\ &= (1-\alpha_i)\boldsymbol{\varepsilon}_i + \alpha_i \sum_j \tilde{A}_{ij}\boldsymbol{\varepsilon}_j + \alpha_i \sum_j \tilde{A}_{ij}(\mathbf{c}_j^\dagger - \mathbf{c}_i^\dagger).\end{aligned} \tag{35}$$

Taking the ℓ_2 norm and applying the triangle inequality yield

$$\begin{aligned}\|\tilde{\mathbf{c}}_i - \mathbf{c}_i^\dagger\|_2 &\leq (1-\alpha_i)\|\boldsymbol{\varepsilon}_i\|_2 + \alpha_i \left\| \sum_j \tilde{A}_{ij}\boldsymbol{\varepsilon}_j \right\|_2 + \alpha_i\eta_i \\ &\leq (1-\alpha_i)\|\boldsymbol{\varepsilon}_i\|_2 + \alpha_i \sum_j \tilde{A}_{ij}\|\boldsymbol{\varepsilon}_j\|_2 + \alpha_i\eta_i,\end{aligned} \tag{36}$$

where the last step follows from the row-stochasticity of $\tilde{A}$. This proves Eq. (34).

Proposition 3 characterizes the trade-off in context calibration. The final context error consists of three components: (i) local estimation error, (ii) propagated neighbor error, and (iii) neighborhood mismatch.

The mixing weight α_i controls this trade-off. Increasing α_i can reduce the local error contribution, but it also increases exposure to neighbor-induced error and neighborhood mismatch. Therefore, a fixed propagation strength cannot adapt to node-specific reliability: reliable nodes should suppress neighbor support, whereas uncertain nodes may benefit from it.

This directly motivates the reliability-guided calibration in REACT, where uncertainty determines how much neighborhood support is introduced.

B.4 Gated Context Modulation Ensures Stable Prediction Updates

Finally, REACT uses gated context modulation to inject the calibrated context into the graph-free predictive representation. Let $\mathbf{h}_i \in \mathbb{R}^{d_h}$ be the base hidden representation of node i, and define its normalized form as $\mathbf{v}_i := \mathrm{LN}(\mathbf{h}_i)$. Given a pre-normalization context $\mathbf{c}$, REACT computes the FiLM-style residual term

$$\mathbf{q}_i(\mathbf{c}) := \mathbf{v}_i \odot \tanh(\boldsymbol{\gamma}(\mathbf{c})) + \boldsymbol{\beta}(\mathbf{c}), \tag{37}$$

where $(\boldsymbol{\gamma}(\mathbf{c}), \boldsymbol{\beta}(\mathbf{c})) := \phi_a(\mathrm{LN}(\mathbf{c}))$, as in Eq. (11). It also computes a modulation gate

$$\mathbf{g}_i(\mathbf{c}) := \mathrm{sigmoid}\left(f_g([\mathbf{v}_i, \mathrm{LN}(\mathbf{c})])\right) \in [0,1]^{d_h}. \tag{38}$$

The modulation residual is

$$\mathbf{r}_i(\mathbf{c}) := \mathbf{g}_i(\mathbf{c}) \odot \mathbf{q}_i(\mathbf{c}). \tag{39}$$

Let $\mathcal{H}$ denote the remaining prediction map after residual injection, including the skip connection with $\mathbf{h}_i$, the output layer normalization, and the final prediction head.

Assumption 1 (Lipschitz Context-to-Modulation Map). For any node i and any contexts $\mathbf{c}, \mathbf{c}'$ in the admissible context domain, the following conditions hold:

$$\|\mathbf{q}_i(\mathbf{c})\|_\infty \le B_q, \qquad \|\mathbf{q}_i(\mathbf{c}) - \mathbf{q}_i(\mathbf{c}')\|_2 \le L_q \|\mathbf{c} - \mathbf{c}'\|_2, \tag{40}$$

and

$$\|\mathbf{g}_i(\mathbf{c}) - \mathbf{g}_i(\mathbf{c}')\|_2 \le L_g \|\mathbf{c} - \mathbf{c}'\|_2. \tag{41}$$

Here L_q and L_g include the effect of context layer normalization with a fixed positive stabilizer.

Assumption 2 (Lipschitz Readout). The prediction map $\mathcal{H}$ is L_h-Lipschitz on the admissible hidden-state domain:

$$\|\mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}')\|_2 \le L_h \|\mathbf{z} - \mathbf{z}'\|_2. \tag{42}$$

Proposition 4 (Prediction Perturbation Bound under Context Error). Under Assumptions 1 and 2, for any contexts $\mathbf{c}$ and $\mathbf{c}'$, the prediction difference satisfies

$$\|\hat{\mathbf{y}}_i(\mathbf{c}) - \hat{\mathbf{y}}_i(\mathbf{c}')\|_2 \le L_{\hat{y}} \|\mathbf{c} - \mathbf{c}'\|_2, \tag{43}$$

where

$$L_{\hat{y}} := L_h(B_q L_g + L_q). \tag{44}$$

Consequently, for the pre-normalization calibrated context $\tilde{\mathbf{c}}_i$ and the ideal latent context $\mathbf{c}_i^\dagger$,

$$\|\hat{\mathbf{y}}_i(\tilde{\mathbf{c}}_i) - \hat{\mathbf{y}}_i(\mathbf{c}_i^\dagger)\|_2 \le L_{\hat{y}} \left[(1-\alpha_i)\|\boldsymbol{\varepsilon}_i\|_2 + \alpha_i \sum_j \tilde{A}_{ij} \|\boldsymbol{\varepsilon}_j\|_2 + \alpha_i \eta_i \right]. \tag{45}$$

Proof. For fixed node i, the residual difference can be decomposed as

$$\begin{aligned} \mathbf{r}_i(\mathbf{c}) - \mathbf{r}_i(\mathbf{c}') &= \mathbf{g}_i(\mathbf{c}) \odot \mathbf{q}_i(\mathbf{c}) - \mathbf{g}_i(\mathbf{c}') \odot \mathbf{q}_i(\mathbf{c}') \\ &= \big(\mathbf{g}_i(\mathbf{c}) - \mathbf{g}_i(\mathbf{c}')\big) \odot \mathbf{q}_i(\mathbf{c}) + \mathbf{g}_i(\mathbf{c}') \odot \big(\mathbf{q}_i(\mathbf{c}) - \mathbf{q}_i(\mathbf{c}')\big). \end{aligned} \tag{46}$$

Taking norms and using $\|\mathbf{a} \odot \mathbf{b}\|_2 \le \|\mathbf{a}\|_2 \|\mathbf{b}\|_\infty$ together with $\mathbf{g}_i(\mathbf{c}') \in [0,1]^{d_h}$, we obtain

$$\begin{aligned} \|\mathbf{r}_i(\mathbf{c}) - \mathbf{r}_i(\mathbf{c}')\|_2 &\le \|\mathbf{g}_i(\mathbf{c}) - \mathbf{g}_i(\mathbf{c}')\|_2 \|\mathbf{q}_i(\mathbf{c})\|_\infty + \|\mathbf{q}_i(\mathbf{c}) - \mathbf{q}_i(\mathbf{c}')\|_2 \\ &\le (B_q L_g + L_q)\|\mathbf{c} - \mathbf{c}'\|_2. \end{aligned} \tag{47}$$

Since $\mathcal{H}$ is L_h-Lipschitz, the prediction satisfies

$$\|\hat{\mathbf{y}}_i(\mathbf{c}) - \hat{\mathbf{y}}_i(\mathbf{c}')\|_2 \le L_h \|\mathbf{r}_i(\mathbf{c}) - \mathbf{r}_i(\mathbf{c}')\|_2 \le L_h(B_q L_g + L_q)\|\mathbf{c} - \mathbf{c}'\|_2. \tag{48}$$

This proves Eq. (43). The combined bound in Eq. (45) follows by setting $\mathbf{c} = \tilde{\mathbf{c}}_i$, $\mathbf{c}' = \mathbf{c}_i^\dagger$, and applying Proposition 3.

Proposition 4 shows that the prediction perturbation induced by context is Lipschitz-bounded. This means that errors in context estimation lead to controlled and proportional changes in prediction, rather than unbounded deviations.

This property is critical under OOD settings: even when context is imperfect, REACT avoids unstable prediction behavior by constraining how context affects the predictive representation.

B.5 Computational Complexity

REACT's graph-dependent cost is linear in the number of sparse support edges. Let B be the batch size, N the number of nodes, d_c the context dimension, and k the average degree of the sparse support graph. The context aggregation in Eq. (10) is a sparse matrix–context multiplication with $|\mathcal{E}_s| \approx Nk$ nonzero support entries, and therefore costs

$$O(B|\mathcal{E}_s|d_c) = O(BNkd_c). \tag{49}$$

The corresponding graph storage is $O(Nk)$, apart from the node-level context activations of size $O(BNd_c)$.

All other components are shared node-wise transformations. The graph-free encoder applies L residual linear channel-mixing blocks independently to each node, with cost $O(BNLd_h^2)$ for hidden width d_h, while the context estimator, uncertainty router, gated modulation module, and prediction head are lightweight per-node projections or MLPs. These terms scale linearly with B and N and do not depend on the support degree k. Thus, increasing the support graph affects only the sparse context-calibration step, whereas the predictive path remains graph-free. REACT therefore achieves robustness through selective structural support, while avoiding the heavy and repeated graph propagation cost of conventional spatio-temporal GNNs.

C Experimental Setup

This appendix presents the experimental setup in a structured manner. Section C.1 introduces datasets and evaluation protocols, Section C.2 summarizes baselines, Section C.3 provides implementation details, and Section C.4 defines evaluation metrics.

C.1 Evaluation Protocols and Datasets

We evaluate REACT on ten real-world spatio-temporal datasets under complementary OOD protocols. The benchmark includes four LargeST road-network datasets (SD, GBA, GLA, CA) and six urban datasets covering pedestrian flow, taxi demand, bike sharing, traffic speed, traffic flow, and 311 service requests. Table 4 summarizes dataset statistics, including temporal splits, graph sizes, and structural changes.

Data sources and licenses. The four LargeST datasets are taken from the LargeST benchmark [21], which is built on traffic data from the Caltrans Performance Measurement System (PeMS) and released under the CC BY-NC 4.0 license. The six urban datasets are obtained from the ST-OOD benchmark [32] and originate from public data providers: PedZurich from the automatic pedestrian counts of the City of Zurich (Open Data Zürich, CC0); TaxiCHI from the City of Chicago Data Portal (Terms of Use); BikeCHI from the Divvy bike-sharing system data (Divvy Data License Agreement); SpeedNYC (NYC DOT traffic speeds) and 311NYC (311 Service Requests) from NYC Open Data (Terms of Use); and FlowSC from Caltrans PeMS (PEMS08). We use all datasets solely for non-commercial research, in accordance with these licenses and terms.

OOD evaluation protocols. We consider two complementary OOD settings that isolate temporal and structural shifts.

❶ **Temporal shift.** Models are trained and validated on earlier time periods and evaluated on later periods. For LargeST datasets, we use the first 60% of 2017 for training, the next 20% for validation, and test on later-year data (2018–2021). For the six urban datasets, we use the first 60% of the first year for training, the next 20% for validation, and the final 20% of the same year for testing (Table 4). This protocol induces temporal distribution drift between training and testing.

❷ **Structural shift.** For fixed-graph datasets, we construct node-level OOD splits. Models are trained on a base graph, while test graphs simulate real deployment changes by (i) removing 10% of training nodes (node disappearance), and (ii) introducing 30% unseen nodes (graph expansion). This protocol reflects practical scenarios where sensor availability or service regions evolve over time.

Characterizing temporal OOD. To quantify temporal shift in LargeST, we compare within-year variation (2017) and cross-year differences (2018–2021) using two complementary metrics: (i)

Table 4: Dataset statistics and OOD evaluation protocols.

Dataset	Train/Val set		Test set		
	Time range	Graph Nodes	Test period	Structural shift	
				New nodes	Removed nodes
SD	01/01–10/19 in 2017	550	10/20–12/31 in 2018–2021	165	55
GBA	01/01–10/19 in 2017	1809	10/20–12/31 in 2018–2021	542	180
GLA	01/01–10/19 in 2017	2949	10/20–12/31 in 2018–2021	884	294
CA	01/01–10/19 in 2017	6615	10/20–12/31 in 2018–2021	1984	661
PedZurich	01/01–10/19 in 2019	76	10/20–12/31 in 2019	22	7
TaxiCHI	01/01–10/19 in 2013	59	10/20–12/31 in 2013	17	5
BikeCHI	01/01–10/19 in 2019	468	10/20–12/31 in 2019	140	46
SpeedNYC	03/01–05/13 in 2019	106	05/13–05/31 in 2019	31	10
FlowSC	07/01–08/19 in 2016	130	08/19–08/31 in 2016	39	13
311NYC	01/01–10/19 in 2019	54	10/20–12/31 in 2019	16	5

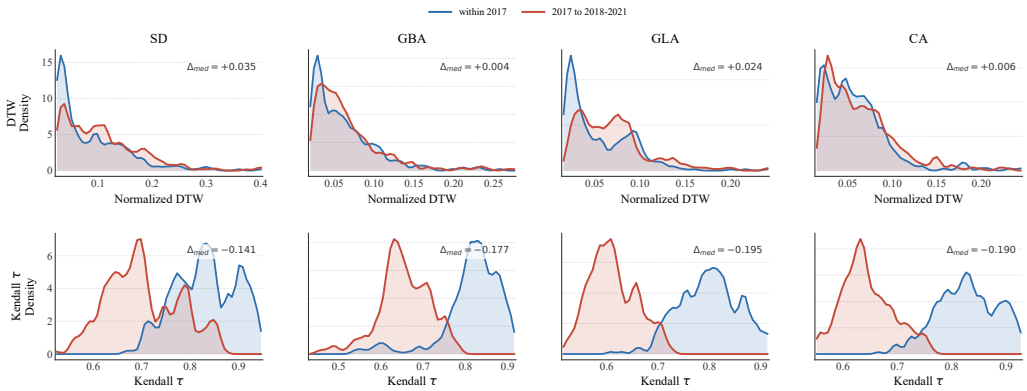

Figure 7: Distribution-level characterization of temporal OOD. Columns correspond to datasets, and rows compare daily temporal profiles (DTW) and spatial rankings (Kendall τ). Blue curves denote within-2017 variation, and red curves denote cross-year differences (2018–2021 vs. 2017).

normalized Dynamic Time Warping (DTW) for temporal profile similarity, and (ii) Kendall's τ for spatial rank consistency across nodes.

Figure 7 shows that cross-year samples consistently deviate from within-2017 variation, but the shift manifests differently across metrics. DTW reveals dataset-dependent changes in temporal profiles, while Kendall's τ shows a more consistent and pronounced degradation, indicating substantial drift in node-wise relative behavior.

These observations suggest that temporal OOD is not merely a shift in marginal time-series patterns, but also reflects evolving spatial dependencies. This motivates models that remain robust to both temporal variation and structural evolution.

C.2 Baseline Families and OOD Adaptation

We compare REACT against two baseline families: IID baselines and OOD-specific methods. All models are evaluated under identical data splits, prediction horizons, and metrics.

To ensure fair evaluation under node-level OOD, we adapt IID baselines by removing graph-specific components that assume fixed node identities (e.g., static node embeddings and adaptive graph learning modules). This avoids invalid indexing and information leakage when evaluating unseen nodes.

IID baselines.

- **STGCN** [38]: combines graph convolution and temporal convolution to model spatio-temporal dependencies.
- **GWNet** [35]: integrates adaptive graph convolution with dilated temporal convolution.
- **D2STGNN** [28]: decouples dynamic spatial and temporal dependencies.
- **STID** [27]: uses identity embeddings for compact spatio-temporal modeling.
- **BigST** [13]: scalable graph model with linear-complexity spatial modeling.
- **RPMixer** [37]: mixer-style architecture with random projection-based efficiency.

OOD-specific baselines.

- **CaST** [36]: causal adjustment for stable prediction under shift.
- **CauSTG** [45]: invariant relation learning across environments.
- **STONE** [31]: semantic graph construction with perturbation-based invariance.
- **STEVE** [16]: confounder-aware representation learning.
- **STOP** [23]: centralized interaction modeling for robustness.

C.3 Implementation Details

All baselines follow their original hyperparameter settings unless modification is required for node-shift evaluation. Experiments are conducted on a Rocky Linux 8.8 server with NVIDIA A40 GPUs.

REACT is implemented in Python 3.8.20 with PyTorch 2.0.1. We use the Adam optimizer with an initial learning rate of 0.001, decay the learning rate by 0.5 every 5 epochs, and apply early stopping with a patience of 10 based on validation loss.

C.4 Metric Definitions

We report both accuracy and efficiency metrics to evaluate predictive quality and computational cost.

Accuracy metrics. We use MAE, RMSE, and MAPE:

$$\mathrm{MAE} = \frac{1}{M}\sum_{k=1}^{M}|y_k - \hat{y}_k|, \quad \mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(y_k - \hat{y}_k)^2}, \quad \mathrm{MAPE} = \frac{100\%}{M}\sum_{k=1}^{M}\left|\frac{y_k - \hat{y}_k}{y_k}\right|,$$

where y_k and $\hat{y}_k$ are the ground truth and prediction of the k-th evaluated entry, and M is the number of evaluated entries over test samples, horizon steps, and nodes.

Efficiency metrics. We report Latency, Epoch time, FLOPs, and Memory usage. Latency measures test-time inference cost, Epoch time reflects training efficiency, FLOPs quantify computational complexity, and Memory records peak GPU usage. Lower values indicate better efficiency across all metrics.

D Extended Experimental Results

This section complements the main text with detailed results that further validate the robustness, efficiency, and mechanism of REACT. We organize the experiments from direct support of the main claims to controlled diagnostics. Section D.1 reports full benchmark tables, Section D.2 isolates temporal and spatial OOD effects, Section D.3 checks baseline fairness under complete and missing static metadata, and Section D.4 analyzes hyperparameter sensitivity.

D.1 Complete Results

This subsection expands RQ1 with year-wise results on the four LargeST datasets. Tables 5 and 6 report MAE, RMSE, and MAPE at Horizons 3, 6, and 12 and their average for the 2018 and 2019 test years, respectively. Across the 2018 and 2019 settings, REACT achieves the lowest average MAE and the lowest Horizon-12 MAE in all eight dataset-year cases, showing that the reliability-guided context calibration remains consistently effective before the more irregular 2020 traffic regime.

Table 5: Performance comparison on four LargeST datasets in 2018. MAPE is reported in percentage. Best results are **bold**, second-best are underlined.

Dataset	Method	Horizon 3			Horizon 6			Horizon 12			Average		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SD	STGCN	18.88	29.53	12.66%	24.74	38.15	16.41%	34.97	54.05	23.13%	25.22	38.95	16.76%
	GWNet	22.03	33.73	15.59%	29.85	45.60	21.23%	41.20	63.43	31.13%	29.92	45.86	21.68%
	D2STGNN	18.81	29.33	12.72%	24.61	38.11	16.55%	33.74	52.09	23.86%	24.80	38.35	17.15%
	STID	18.56	29.54	**11.33%**	25.59	40.48	15.77%	37.77	59.33	24.32%	26.20	41.33	16.39%
	BigST	19.19	30.69	12.09%	26.52	41.64	16.72%	39.36	60.02	26.09%	27.14	42.23	17.40%
	RPMixer	21.04	34.19	12.48%	31.32	51.02	19.25%	50.45	80.33	34.60%	32.64	52.59	20.86%
	CaST	24.48	35.43	30.14%	30.81	45.78	31.76%	43.24	64.73	38.11%	31.68	46.78	32.60%
	CauSTG	19.63	30.50	12.98%	26.33	40.59	17.52%	36.45	56.44	26.01%	26.44	40.83	18.07%
	STONE	19.35	29.89	17.52%	25.55	39.49	21.53%	35.74	56.33	29.10%	25.92	40.28	22.07%
	STEVE	19.21	29.97	12.51%	25.36	39.04	16.57%	35.66	55.06	24.74%	25.75	39.70	17.12%
	STOP	17.85	28.20	12.25%	23.82	37.85	16.90%	33.44	53.75	25.72%	24.15	38.45	17.57%
	REACT	**16.71**	**26.59**	11.75%	**20.70**	**32.99**	**15.04%**	**26.99**	**44.37**	**19.64%**	**20.83**	**33.49**	**15.05%**
GBA	STGCN	21.90	35.10	20.00%	28.75	44.62	28.06%	39.67	60.16	41.44%	29.07	45.11	28.79%
	GWNet	23.86	36.79	21.29%	31.58	46.96	29.83%	42.11	61.22	44.75%	31.40	46.76	30.60%
	D2STGNN	21.04	33.44	19.93%	27.12	41.62	27.52%	36.09	54.61	38.71%	27.23	41.93	27.68%
	STID	21.60	35.05	20.56%	29.54	45.96	30.49%	42.22	64.13	47.43%	29.99	46.72	31.58%
	BigST	22.40	36.45	18.67%	30.60	47.31	28.52%	43.27	64.66	44.10%	30.93	47.76	29.23%
	RPMixer	23.89	39.47	19.33%	34.68	55.78	31.59%	54.43	85.66	54.79%	35.96	57.73	33.36%
	CaST	27.19	39.91	38.08%	34.43	50.55	43.84%	47.48	69.80	55.07%	35.31	51.73	44.83%
	CauSTG	22.04	34.72	20.36%	29.05	44.02	28.40%	38.77	57.20	42.78%	28.95	43.90	29.08%
	STONE	22.39	34.77	19.35%	28.20	42.85	28.09%	38.51	57.86	41.79%	28.75	43.75	28.80%
	STEVE	22.16	35.11	21.55%	29.08	44.59	30.13%	40.30	60.71	44.14%	29.46	45.24	30.79%
	STOP	20.56	33.46	20.55%	26.56	41.88	27.79%	35.31	54.91	38.76%	26.61	42.10	27.94%
	REACT	**19.67**	**32.08**	**17.88%**	**24.24**	**38.59**	**24.04%**	**30.99**	**48.94**	**33.21%**	**24.25**	**38.75**	**24.20%**
GLA	STGCN	21.99	34.11	17.81%	29.85	45.93	21.75%	42.17	63.80	30.98%	30.23	46.30	22.67%
	GWNet	23.73	36.25	15.43%	33.25	49.78	21.95%	45.61	66.81	33.39%	33.00	49.22	22.61%
	D2STGNN	–	–	–	–	–	–	–	–	–	–	–	–
	STID	21.07	33.45	15.28%	29.67	46.24	21.76%	42.74	65.32	31.26%	30.05	46.68	22.03%
	BigST	21.75	34.44	13.84%	30.65	47.05	20.58%	44.35	65.78	31.96%	30.99	47.34	21.03%
	RPMixer	23.32	38.08	13.08%	34.84	56.90	21.34%	54.49	87.43	39.01%	35.91	58.32	23.06%
	CaST	28.20	40.76	34.59%	37.07	54.21	37.77%	57.02	82.32	49.01%	39.66	57.35	39.76%
	CauSTG	21.37	33.21	13.86%	29.39	44.66	19.28%	41.03	60.92	29.15%	29.50	44.68	19.94%
	STONE	–	–	–	–	–	–	–	–	–	–	–	–
	STEVE	22.52	35.81	31.44%	30.40	47.28	31.56%	42.50	64.67	36.22%	30.67	47.55	32.63%
	STOP	20.48	32.02	14.23%	28.02	42.95	21.98%	39.24	60.08	35.21%	28.24	43.47	22.84%
	REACT	**18.79**	**29.80**	**11.21%**	**23.82**	**37.27**	**14.92%**	**31.24**	**49.17**	**20.87%**	**23.86**	**37.51**	**15.12%**
CA	STGCN	20.62	32.88	21.14%	28.08	43.79	28.38%	39.99	61.28	38.18%	28.51	44.40	28.55%
	GWNet	21.95	34.82	21.86%	30.67	47.22	29.71%	42.60	63.78	42.84%	30.57	46.91	30.16%
	D2STGNN	–	–	–	–	–	–	–	–	–	–	–	–
	STID	20.45	33.08	27.66%	28.99	45.83	39.64%	41.90	64.82	49.62%	29.34	46.24	38.18%
	BigST	20.59	33.67	15.39%	28.85	45.52	22.57%	41.97	64.11	36.02%	29.32	46.03	23.51%
	RPMixer	21.88	37.07	15.10%	32.76	55.20	24.68%	51.66	85.53	43.35%	33.87	56.82	26.23%
	CaST	26.61	38.88	45.59%	34.33	51.00	51.86%	48.65	71.59	60.68%	35.43	52.10	52.21%
	CauSTG	20.68	33.02	22.27%	28.52	44.36	29.63%	39.89	60.71	41.10%	28.66	44.47	30.41%
	STONE	–	–	–	–	–	–	–	–	–	–	–	–
	STEVE	22.37	40.57	54.04%	29.79	50.52	47.86%	40.80	64.61	46.08%	29.97	50.54	49.39%
	STOP	20.26	31.99	20.95%	28.13	43.04	31.28%	39.22	58.56	44.60%	28.12	42.92	30.96%
	REACT	**18.39**	**29.67**	**14.82%**	**23.73**	**37.33**	**20.45%**	**31.74**	**49.47**	**28.52%**	**23.81**	**37.54**	**20.52%**

D.2 Isolated-Shift Robustness

We further disentangle joint OOD into spatial-only (SOOD) and temporal-only (TOOD) settings to isolate the source of robustness. In SOOD, the temporal regime remains within 2017 while the graph structure changes via a spatial shift ratio r_s. In TOOD, the graph is fixed, and models are evaluated for future years. We exclude 2020 in TOOD due to its atypical external disruption.

Table 7 supports two key findings. First, REACT achieves the lowest SOOD MAE in nearly all dataset-ratio settings, with SD at $r_s = 0.3$ as the only exception where it remains second-best, demonstrating strong robustness to unseen nodes and graph expansion. Second, after excluding 2020, REACT achieves the lowest TOOD MAE in most target years, with SD-2018 as the only exception (19.87 vs. 19.50), where it remains second-best. These results indicate that REACT improves robustness along both spatial and temporal dimensions, rather than specializing in a single type of shift.

Table 6: Performance comparison on four LargeST datasets in 2019. MAPE is reported in percentage. Best results are **bold**, second-best are underlined.

Dataset	Method	Horizon 3			Horizon 6			Horizon 12			Average		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SD	STGCN	19.75	31.41	14.14%	25.94	40.55	18.27%	36.46	56.80	25.77%	26.37	41.24	18.68%
	GWNet	23.46	36.05	17.21%	31.87	48.57	23.52%	43.95	66.90	34.89%	31.87	48.61	24.08%
	D2STGNN	20.45	32.14	14.77%	27.26	42.23	19.65%	37.70	58.04	28.69%	27.43	42.43	20.35%
	STID	19.22	31.10	**12.24%**	26.48	42.33	17.12%	38.82	61.22	26.31%	27.04	43.06	17.76%
	BigST	19.56	31.83	13.91%	26.69	42.57	17.72%	38.81	60.48	27.18%	27.18	43.05	18.36%
	RPMixer	21.68	35.73	13.50%	32.34	53.23	21.04%	51.68	82.54	37.44%	33.59	54.58	22.68%
	CaST	24.83	36.28	32.57%	31.27	46.82	34.12%	43.65	65.36	40.79%	32.09	47.66	35.05%
	CauSTG	20.91	32.87	14.61%	28.24	43.73	19.97%	38.98	59.93	29.78%	28.25	43.70	20.55%
	STONE	20.04	31.49	20.81%	26.33	41.21	25.84%	36.67	57.70	34.40%	26.70	41.85	26.26%
	STEVE	20.33	32.18	13.97%	27.03	42.08	18.60%	37.85	58.68	27.62%	27.36	42.59	19.19%
	STOP	18.60	29.80	13.87%	24.88	39.95	19.13%	34.80	56.30	28.90%	25.15	40.44	19.81%
	REACT	**17.50**	**28.31**	13.09%	**21.86**	**35.42**	**17.01%**	**28.48**	**47.29**	**22.73%**	**21.94**	**35.81**	**17.14%**
GBA	STGCN	21.85	34.75	21.53%	29.36	45.40	29.21%	41.17	62.55	41.65%	29.72	46.03	29.79%
	GWNet	23.97	36.36	21.48%	32.32	47.35	30.02%	43.53	62.73	44.60%	32.08	47.12	30.67%
	D2STGNN	21.43	33.26	20.49%	28.39	42.64	28.23%	38.52	57.36	39.12%	28.49	42.96	28.25%
	STID	21.19	34.06	21.49%	29.51	45.51	29.84%	42.71	64.62	45.58%	29.97	46.30	31.12%
	BigST	21.60	34.72	17.42%	30.04	45.83	27.29%	43.38	64.29	42.82%	30.45	46.42	28.00%
	RPMixer	23.25	37.69	17.75%	34.44	53.78	29.23%	54.75	82.87	51.90%	35.77	55.55	31.18%
	CaST	26.71	38.71	37.12%	33.86	49.34	42.52%	46.96	69.08	53.24%	34.81	50.68	43.50%
	CauSTG	22.11	34.19	21.13%	29.78	44.34	28.86%	40.38	58.95	42.36%	29.68	44.29	29.34%
	STONE	23.50	36.74	23.43%	30.13	45.80	34.10%	41.63	62.17	51.60%	30.72	46.74	35.22%
	STEVE	21.83	34.17	22.02%	29.32	44.60	30.28%	41.35	62.18	43.89%	29.74	45.32	30.94%
	STOP	20.99	33.34	21.46%	28.09	43.26	29.27%	38.32	58.07	40.88%	28.16	43.48	29.36%
	REACT	**19.91**	**31.54**	**16.06%**	**25.23**	**38.86**	**21.70%**	**32.67**	**49.70**	**30.06%**	**25.15**	**38.83**	**21.83%**
GLA	STGCN	21.56	33.43	15.60%	29.57	45.26	19.89%	42.03	63.12	29.77%	29.95	45.64	20.87%
	GWNet	23.65	36.18	15.05%	33.42	50.19	21.75%	45.94	67.40	33.50%	33.11	49.51	22.43%
	D2STGNN	–	–	–	–	–	–	–	–	–	–	–	–
	STID	20.68	32.76	13.22%	29.40	45.41	19.40%	42.66	64.90	29.76%	29.80	46.02	20.03%
	BigST	21.14	33.61	13.59%	29.75	46.03	20.23%	42.93	64.54	31.30%	30.07	46.34	20.67%
	RPMixer	23.08	37.48	13.50%	35.00	56.29	22.48%	55.26	86.65	42.01%	36.11	57.63	24.44%
	CaST	27.82	40.22	34.96%	36.14	53.07	37.60%	55.44	80.27	48.87%	38.76	56.18	39.78%
	CauSTG	21.25	32.98	13.19%	29.57	44.91	18.90%	41.43	61.42	29.00%	29.64	44.84	19.50%
	STONE	–	–	–	–	–	–	–	–	–	–	–	–
	STEVE	21.40	33.27	20.31%	29.28	44.68	23.77%	41.43	62.20	32.54%	29.61	45.06	24.85%
	STOP	19.71	30.86	13.01%	26.83	41.26	20.01%	37.38	57.56	31.30%	27.02	41.77	20.55%
	REACT	**19.17**	**29.96**	**12.41%**	**24.89**	**38.20**	**16.83%**	**33.05**	**50.94**	**23.93%**	**24.88**	**38.41**	**17.08%**
CA	STGCN	20.42	33.22	31.46%	28.07	44.47	36.52%	40.22	62.25	42.84%	28.53	45.09	36.46%
	GWNet	21.84	34.59	24.30%	30.89	47.49	32.71%	43.23	64.60	46.89%	30.78	47.15	33.17%
	D2STGNN	–	–	–	–	–	–	–	–	–	–	–	–
	STID	20.36	33.01	37.82%	29.26	46.37	51.40%	42.37	65.52	58.92%	29.55	46.62	48.72%
	BigST	19.97	32.52	15.55%	28.09	44.27	22.92%	41.33	63.18	37.21%	28.62	44.86	23.99%
	RPMixer	22.05	36.84	15.34%	32.95	54.49	24.85%	51.84	83.99	44.09%	34.06	56.02	26.59%
	CaST	26.39	38.13	49.03%	33.94	50.05	55.60%	48.04	70.51	64.57%	35.05	51.20	55.89%
	CauSTG	20.57	32.74	27.14%	28.69	44.36	35.07%	40.32	61.01	47.44%	28.81	44.48	36.04%
	STONE	–	–	–	–	–	–	–	–	–	–	–	–
	STEVE	23.05	44.46	72.35%	30.53	53.65	60.94%	41.53	66.74	54.35%	30.70	53.71	63.13%
	STOP	20.16	31.62	23.99%	28.37	43.15	35.50%	39.81	59.10	49.17%	28.33	42.99	34.79%
	REACT	**18.28**	**29.14**	**15.30%**	**24.01**	**37.24**	**21.41%**	**32.37**	**49.70**	**30.06%**	**24.04**	**37.38**	**21.47%**

D.3 Static Metadata and Baseline Fairness

REACT uses only the per-node metadata released with each dataset: coordinates and lane count (z-scored), road id, direction, district and segment type (one-hot), and node degree from the predefined adjacency. Identifier columns are discarded, and every normalization statistic and one-hot vocabulary is fitted on training nodes only, so no node-identity or test-node information enters the model. To keep the comparison fair, we re-train a set of representative baselines with the same feature vector by channel-wise concatenation. Both tables below report All MAE and use these re-trained baselines.

Baselines with metadata. Table 8(a) evaluates models trained on 2017 and frozen at test time on 2018–2021, with 30% of nodes added and 10% removed. Even when these representative baselines see complete metadata, REACT attains the lowest MAE on all four datasets.

Table 7: SOOD and TOOD results with all OOD baselines. SOOD metrics are computed on unseen (UnF) nodes, and TOOD metrics on all nodes. '–' indicates out-of-memory. Best and second-best results for each metric within each setting are marked as **bold** and underlined.

Dataset	Method	SOOD: spatial shift only (UnF nodes)						TOOD: temporal shift only (All nodes)					
		$r_s = 0.1$		$r_s = 0.2$		$r_s = 0.3$		2018		2019		2021	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SD	CaST	31.35	47.44	31.32	47.44	31.20	47.35	31.92	48.30	32.18	48.95	31.54	47.91
	CauSTG	23.15	**36.33**	25.52	39.51	26.56	41.10	**19.50**	**31.03**	21.22	**34.17**	26.44	39.67
	STONE	24.07	37.37	24.49	38.06	25.19	39.17	28.24	43.94	28.94	45.27	30.43	46.83
	STEVE	24.39	37.45	24.80	38.16	24.88	38.36	24.27	37.54	25.75	40.14	29.38	44.68
	STOP	23.28	37.00	23.29	**37.09**	**23.16**	**36.96**	23.65	37.87	24.34	39.31	26.28	40.79
	REACT	**22.42**	36.39	**23.10**	37.64	23.60	37.77	19.87	31.76	**21.18**	34.32	**23.83**	**37.85**
GBA	CaST	36.78	52.97	36.71	52.90	36.64	52.81	39.48	55.12	39.29	54.47	35.00	49.00
	CauSTG	24.58	37.69	25.80	39.23	26.97	40.74	24.78	38.80	26.01	39.62	25.12	**38.17**
	STONE	28.09	42.72	28.31	43.08	28.75	43.75	34.55	54.26	34.97	54.16	31.08	48.46
	STEVE	27.03	40.93	27.10	41.07	27.17	41.21	28.83	44.32	29.52	44.94	26.68	40.93
	STOP	24.82	39.30	24.75	39.23	24.76	39.26	26.50	42.01	27.93	43.11	25.19	39.48
	REACT	**23.06**	**35.91**	**23.10**	**36.07**	**23.34**	**36.55**	**23.76**	**38.19**	**24.82**	**38.52**	**25.01**	38.49
GLA	CaST	39.19	57.45	39.28	57.63	39.17	57.48	39.24	57.24	38.61	56.15	36.14	52.60
	CauSTG	26.01	40.65	26.68	41.69	27.15	42.30	27.23	42.00	27.73	42.35	27.70	42.50
	STONE	–	–	–	–	–	–	–	–	–	–	–	–
	STEVE	28.54	44.30	28.71	44.47	28.25	43.84	29.18	45.09	29.46	45.58	28.73	44.39
	STOP	24.31	39.50	24.60	39.78	24.40	39.47	26.40	41.30	26.74	41.40	28.17	43.36
	REACT	**23.61**	**37.94**	**23.79**	**37.95**	**23.99**	**38.12**	**23.28**	**36.79**	**24.45**	**37.89**	**25.68**	**39.69**
CA	CaST	32.30	49.19	32.24	49.38	34.70	52.15	32.32	49.00	31.82	48.02	28.95	43.54
	CauSTG	28.26	44.32	28.13	44.23	27.81	43.80	28.00	43.69	27.66	43.10	25.42	39.70
	STONE	–	–	–	–	–	–	–	–	–	–	–	–
	STEVE	28.19	43.97	28.08	43.99	27.65	43.40	28.80	44.88	28.78	45.02	26.08	40.65
	STOP	24.22	38.90	24.18	39.11	23.77	38.59	28.58	43.64	29.16	44.26	27.03	41.10
	REACT	**22.93**	**36.37**	**23.00**	**36.70**	**23.17**	**36.86**	**23.75**	**37.57**	**23.96**	**37.27**	**23.72**	**37.22**

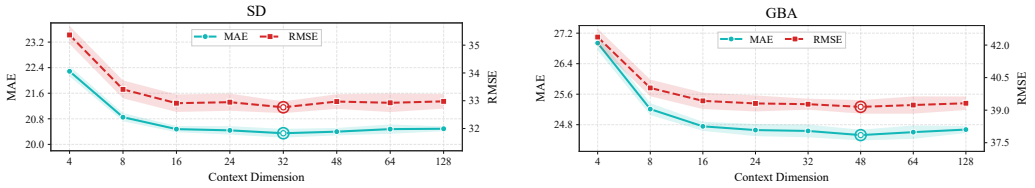

Figure 8: Sensitivity to context dimension on large-scale datasets.

Baselines with missing metadata. Metadata is often incomplete in practice: newly registered sensors may lack records, and third-party feeds may cover only part of a network. Table 8(b) removes the metadata of 50% of test nodes while keeping all weights frozen. REACT is informed through its availability indicator m_i (Eq. 7), whereas baselines have no such input and receive zero-filled vectors. REACT remains the best on every dataset, ahead of the strongest baseline, by 21.07 (SD), 21.28 (GBA), 15.13 (GLA) and 20.19 (CA) MAE.

D.4 Hyperparameter Sensitivity (*RQ6*)

We analyze the sensitivity of REACT to the context dimension on SD and GBA. Since the context dimension directly controls the capacity of context representation and is the most important hyperparameter in REACT, we focus on this parameter for sensitivity analysis. Figure 8 reports the MAE and RMSE under different context dimensions, where the shaded bands denote the standard deviation. Increasing the context dimension from 4 to 16 or 32 consistently reduces both MAE and RMSE,

Table 8: All MAE (lower is better) of baselines re-trained with the same metadata features as REACT. (a) Complete metadata, averaged over 2018–2021 (30% of nodes added, 10% removed). (b) Metadata of 50% of test nodes removed (2017 → 2018, 10% of nodes added, 10% removed). The best result in each column is in **bold**. "–" denotes out-of-memory.

(a) Complete metadata

Method	SD	GBA	GLA	CA
D2STGNN	24.26	26.34	–	–
STONE	24.99	27.88	–	–
GWNet	24.57	27.02	30.11	27.21
RPMixer	28.41	30.64	31.17	27.79
CauSTG	24.58	26.37	27.98	26.64
CaST	35.12	31.16	34.96	33.54
STGCN	25.44	27.13	29.23	26.55
STEVE	25.10	26.86	28.59	25.66
REACT	**23.43**	**25.31**	**25.75**	**24.03**

(b) 50% metadata missing

Method	SD	GBA	GLA	CA
D2STGNN	58.34	54.22	–	–
STONE	85.41	123.00	–	–
GWNet	52.22	63.12	58.25	53.64
RPMixer	52.42	60.27	77.65	127.20
CauSTG	59.19	53.52	51.56	60.77
CaST	76.24	57.25	88.95	60.54
STGCN	83.89	87.90	99.63	88.84
STEVE	43.70	47.45	40.81	45.13
REACT	**22.63**	**26.17**	**25.68**	**24.94**

showing that a very small context space cannot sufficiently capture distributional shifts. After 32, the performance generally becomes stable, indicating that further increasing the dimension mainly introduces redundant capacity. SD achieves its optimum around 32. In contrast, the large-scale GBA dataset performs best at 48, suggesting that more complex scenarios may require slightly larger context capacity. Therefore, we use 32 as the default context dimension for most datasets, while 48 is adopted for GBA.

E More Discussion

E.1 Relation to Retrieval-Augmented and Continual Paradigms

Table 9 separates three paradigms by the information each grants the model at target time: source-only OOD generalization, retrieval-augmented continual generalization, and streaming/continual adaptation. REACT follows the source-only protocol of Definition 2: source knowledge is distilled into fixed parameters $\theta^\star$, and target-time prediction is a direct forward pass with no retrieval, no library, and no updates. STRAP [40], in contrast, retrieves from an explicit pattern library that is continually enriched with patterns extracted from newly arriving graph snapshots, whereas STEV [24], EAC [3], and STBP [19] use newly observed target data to update the predictor or its auxiliary state.

Table 9: Paradigms for spatio-temporal forecasting under distribution shift, distinguished by the information available at target time. $\mathcal{E}_s \subseteq \mathcal{S}$ denotes the source environments, κ indexes sequentially arriving target periods, and w denotes either model parameters or auxiliary adaptation state. †STEV is grouped by scenario rather than as strict continual fine-tuning, since it retrains a variable-scale-agnostic model on pre-expansion data together with limited post-expansion observations.

Paradigm	Example works	Knowledge at target time	Target-time mechanism	Core formulation
Source-only OOD generalization	REACT, STOP [23]	Current input $\mathbf{X}$; historical knowledge is encoded implicitly in $\theta^\star$	Direct prediction with fixed model parameters	Train: $\theta^\star = \arg\min_\theta \max_{e\in\mathcal{E}_s} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim p^e}[\mathcal{L}(f_\theta(\mathbf{X}),\mathbf{Y})]$; Predict: $\hat{\mathbf{Y}} = f_{\theta^\star}(\mathbf{X})$
Retrieval-augmented continual generalization	STRAP [40]	Current input $\mathbf{X}$ and an explicit historical pattern library $\mathcal{M}$	Retrieve relevant patterns and fuse them with the current representation, without task-specific fine-tuning	Retrieve: $\mathbf{r} = \mathrm{Ret}(\mathbf{X},\mathcal{M})$; Predict: $\hat{\mathbf{Y}} = f_{\theta^\star}(\mathbf{X},\mathbf{r})$
Streaming/continual adaptation	STEV† [24], EAC [3], STBP [19]	Newly observed target batches $\mathcal{D}_\kappa = \{(\mathbf{X}^\kappa,\mathbf{Y}^\kappa)\}$ arriving sequentially	Update the model or an auxiliary adaptation state before predicting	$w^\kappa = \arg\min_w \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim p^\kappa}[\mathcal{L}(f_w(\mathbf{X}),\mathbf{Y})]$, initialized from $w^{\kappa-1}$

E.2 Limitations

❶ **Idealized analysis.** Our theory explains the context-stabilization mechanism, but it is not a full generalization guarantee. It relies on latent ideal context, Lipschitz assumptions, and moderate neighborhood mismatch.

❷ **Controlled protocols.** The experiments cover ten traffic and urban datasets under designed temporal and structural OOD splits. Real deployments may include incidents, sensor faults, asynchronous missingness, or changing sensor semantics that are not fully covered here.

❸ **Graph dependence.** REACT assumes that a useful relation graph or proxy graph is available for context calibration. If the graph is very noisy or poorly aligned with predictive similarity, the support module may need extra relation learning.

E.3 Future Work

❶ **Richer context.** Beyond the static node metadata used in this work, we will incorporate external signals such as weather, events, and textual reports to distinguish more types of regime shifts.

❷ **Adaptive relations.** We will replace fixed support with time-varying or reliability-aware relations, which can better support cross-city transfer and partial graph observability.

❸ **Online adaptation.** We will study test-time calibration, online updating, and uncertainty monitoring so that REACT can adjust after deployment while keeping the graph-free backbone stable.

E.4 Broader Impacts

REACT aims to improve the robustness and efficiency of spatio-temporal forecasting systems under temporal and structural distribution shifts. Potential positive impacts include more reliable forecasting for transportation management, urban planning, and public-service monitoring, especially when sensors are added, removed, or deployed under changing demand patterns.

Potential negative impacts mainly stem from erroneous predictions. REACT can still produce inaccurate forecasts under shifts beyond the evaluated protocols, such as abrupt global drops in traffic volume, and such errors may mislead traffic control or resource allocation if forecasts are acted upon without monitoring or human oversight. In addition, forecasting quality depends on sensor coverage and node metadata, so areas with sparse, newly deployed, or poorly documented sensors may receive less accurate forecasts, which could reinforce uneven service quality across regions. We therefore recommend deploying REACT as a decision-support tool with continuous error monitoring and human oversight rather than as a fully automated decision maker.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the paper's scope, the reliability-aware context-calibration contribution, and the empirical claims under spatio-temporal OOD shifts. The claims are supported by the method description, theoretical analysis in Appendix B, and experiments in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Appendix E.2 discusses limitations including the idealized nature of the analysis, controlled OOD protocols, and dependence on a useful relation or proxy graph.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: Appendix B states the assumptions and provides proofs for the propositions on temporal context risk, spatial information degradation, context-calibration error, and prediction perturbation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The paper specifies the model components in Section 4, and the experimental setup in Section 5 and Appendix C, including dataset splits, OOD protocols, baseline adaptation, implementation details, and metrics. The released code provides the executable scripts and instructions for reproducing the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Our source code is available at https://github.com/ZJU-DAILY/REACT.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 5 and Appendix C describe the datasets, OOD construction, metrics, baselines, optimizer, learning-rate schedule, early stopping rule, and hyperparameter sensitivity.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: See Appendix D.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix C.3 reports the hardware and software environment, and Section 5.6 reports latency, per-epoch time, FLOPs, and peak GPU memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The study uses public or properly credited spatio-temporal datasets for research benchmarking and does not involve human-subject experiments, private data collection, or deceptive practices.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes].

Justification: The broader-impact discussion in Appendix E.4 describes potential benefits for robust and efficient urban forecasting, as well as risks from erroneous predictions, uneven sensor coverage, and deployment without monitoring or human oversight.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper does not release a high-risk generative model, scraped unsafe dataset, or dual-use pretrained model. The released assets are forecasting code and experiment scripts for public spatio-temporal benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: The paper credits the original datasets and baseline methods through citations, and Appendix C.1 lists the source and the license or terms of use of each dataset. Baseline methods are cited and described in Section 5 and Appendix C.2.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The released code is documented with environment requirements, data preparation instructions, training and evaluation commands, and reproduction scripts. No new dataset containing human subjects or private information is introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing, participant studies, or other human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The paper does not involve human-subject research, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: The core method does not use LLMs as an important, original, or non-standard component. Any writing, editing, or formatting assistance, if used, does not affect the methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.