ERST: LEARNING INTERACTION FROM ENTITY REPLACEMENT IN SPATIOTEMPORAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatiotemporal forecasting requires modeling interactions among multiple entities whose values change over time. Existing approaches obtain interaction information from entity representations, either by modeling relations among entities or by combining multiple entities into one representation. However, these approaches mainly use the relations or the resulting representation itself and do not directly measure how a representation formed from all entities changes with respect to one entity. We study this change by replacing the representation of one entity with a reference. We propose Entity Replacement for Spatiotemporal Forecasting (ERST), which replaces one entity representation while keeping the other entities, normalization, and function unchanged. ERST applies the same function before and after the replacement and uses the resulting difference as the response for that entity. The reference is constructed from the history of the entity and information from the other entities without directly using the current representation being replaced. The response therefore measures how the representation formed from all entities changes when one entity is replaced. We evaluate ERST on 12 datasets across five domains. ERST improves MAE in 23 of 24 combinations of forecasting architecture and dataset and matches the baseline in the remaining case.

1 INTRODUCTION

Spatiotemporal forecasting aims to predict the future values of systems composed of many interacting entities, such as traffic networks, urban mobility systems, air quality networks, and energy systems. Existing methods model interactions among entities using graphs, attention, or aggregation (Li et al., 2018; Wu et al., 2019; Jiang et al., 2023a; Zaheer et al., 2017; Han et al., 2024b). Methods using graphs or attention construct interaction from relations among entity representations, while methods using aggregation combine multiple entities into one representation.

Aggregation is related to our setting because it forms one representation from all entities (Zaheer et al., 2017; Han et al., 2024b). Using this representation itself as interaction information provides a representation formed from multiple entities, but does not describe how that representation changes when the representation of one entity is replaced. We use the difference obtained before and after replacing one entity representation as interaction information for that entity.

Figure 1 illustrates the distinction between aggregation and ERST on PEMS08. Two observations have a small difference in mean traffic flow, while the values of individual entities differ. Aggregation uses the representation formed from the entities itself as interaction information, which does not describe these differences among entities. This motivates examining how the same function changes when one entity representation is replaced while the other entities remain unchanged. ERST obtains a response for each entity from this change, producing different interac-

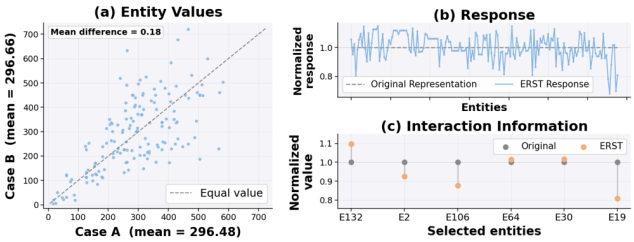


Figure 1: Case study on PEMS08 illustrating the distinction from aggregation. (a) Entity values. (b) Response. (c) Interaction information.

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

tion information for the selected entities. This case illustrates how interaction can be defined from the change associated with each entity rather than from the aggregated representation alone.

Based on this distinction, we propose Entity Replacement for Spatiotemporal Forecasting (ERST). For each entity, ERST replaces only its representation with a reference while keeping the other entities, the number of entities, normalization, and function unchanged. The same function is applied before and after the replacement, and the resulting difference is used as the response for that entity. The key novelty is to use this change for each entity, rather than either representation itself, as interaction information for forecasting.

The distinction of ERST lies in how interaction is defined. Existing aggregation mechanisms use the representation formed from multiple entities itself as interaction information. ERST instead obtains two results from the same function by replacing only one entity representation with a reference. Their difference defines the response for that entity, and this response is used as interaction information for forecasting.

We evaluate ERST across multiple spatiotemporal domains, system scales, and forecasting architectures. The experiments examine whether the response provides information beyond the representation itself, whether replacing one entity while keeping the others unchanged is necessary, and whether the same interaction can be used with different forecasting architectures.

Our contributions are summarized as follows:

- We define interaction for each entity by replacing its representation with a reference and using the difference obtained from the same function before and after the replacement as the response for that entity.
- We propose ERST, which applies this response construction to each entity and uses the resulting response to update its representation for forecasting.
- We evaluate ERST across multiple spatiotemporal domains, system scales, and forecasting architectures, with controlled ablations and experiments using different random seeds.

2 Preliminaries and Related Work

2.1 Problem Definition

Consider a spatiotemporal system with N entities. Given an observation window of length T, we denote the historical observations as $\mathbf{X}_{t-T+1:t} \in \mathbb{R}^{T\times N\times C}$, where C is the number of input features. The forecasting task is to predict the future values over the next K steps, written as $\hat{\mathbf{Y}}_{t+1:t+K} = f_\theta(\mathbf{X}_{t-T+1:t})$, where $\hat{\mathbf{Y}}_{t+1:t+K} \in \mathbb{R}^{K\times N}$ and f_θ denotes a forecasting model.

Let $\mathbf{h}_i$ denote the representation of entity i at the current forecasting stage, and let $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\}$ contain the representations of all entities. Existing methods mainly construct interactions or combine these representations directly from $\mathbf{H}$. In this work, we study how the resulting representation changes when the representation of one entity is replaced by a reference. For an entity i, we use a reference $\mathbf{b}_i$ while keeping the entity and all other representations unchanged, resulting in $\mathbf{H}^{(i)}_{\text{rep}} = \{\mathbf{h}_1, \ldots, \mathbf{b}_i, \ldots, \mathbf{h}_N\}$. The objective is to measure the difference between the representations obtained from $\mathbf{H}$ and $\mathbf{H}^{(i)}_{\text{rep}}$ and to use this response for that entity in forecasting.

2.2 Spatial Interaction Modeling

Spatiotemporal forecasting models capture dependencies among entities through several forms of interaction. Methods based on graphs propagate information through predefined or learned spatial structures (Yu et al., 2018; Li et al., 2018; Wu et al., 2019; Bai et al., 2020; Wu et al., 2020), while methods using attention learn relations among entities from the input (Jiang et al., 2023a). Recent methods also reduce the cost of modeling interactions in large spatial systems (Han et al., 2024a; Zhao et al., 2026). These methods construct interaction from explicit or learned relations among entity representations. ERST instead uses the change in a representation formed from all entities after replacing the representation of one entity with a reference as interaction information.

2.3 Interaction through Aggregation

Aggregation provides an alternative to explicitly modeling relations between pairs of entities. Set models construct one representation by aggregating transformed elements (Zaheer et al., 2017), while Set Transformer uses attention to model interactions among elements (Lee et al., 2019). In forecasting, SOFTS also aggregates multiple series into one representation (Han et al., 2024b). These approaches use the resulting representation as interaction information.

ERST differs from existing aggregation approaches in how interaction is defined from the resulting representation. Existing approaches use the representation formed from multiple entities itself as interaction information. ERST evaluates the same function before and after replacing one entity representation and uses their difference as the response for that entity. Thus, the distinction concerns what is used as interaction information rather than introducing a new aggregation operation.

Our formulation is also related to analyses that modify or remove parts of the input to examine model behavior (Fong & Vedaldi, 2017; Yeh et al., 2019; Covert et al., 2021; Thiagarajan et al., 2021). Prior studies have shown that replacements that ignore dependencies among variables can produce unrealistic inputs or inputs outside the data distribution (Aas et al., 2021; Frye et al., 2021). These approaches use the resulting change mainly for analysis or explanation. ERST instead defines the change between two evaluations of the same function as the interaction itself during forecasting. The same entities, normalization, and function are kept before and after the replacement, which distinguishes this operation from removing an entity. The resulting response is not interpreted as a causal effect. Related causal approaches in spatiotemporal forecasting address distribution shift or dynamic spatial causation (Xia et al., 2023).

3 Proposed Method

3.1 Model Overview

Given the historical observations $\mathbf{X}_{t-T+1:t}$ defined in Section 2.1, ERST first encodes the history of each entity and then applies L interaction layers. Each layer applies a feedforward transformation to each entity followed by the interaction defined by ERST. After the final layer, a decoder applied to each entity produces the future prediction. We denote the representation dimension by D.

For entity i, the input vector at each time step τ is first projected to a scalar representation as $\bar{x}_{\tau,i} = \mathbf{w}_f^\top \mathbf{X}_{\tau,i,:}$, where $\mathbf{w}_f \in \mathbb{R}^C$. The resulting history is collected as $\mathbf{x}_i = [\bar{x}_{t-T+1,i}, \ldots, \bar{x}_{t,i}]^\top \in \mathbb{R}^T$ and mapped to $\mathbf{q}_i = \mathbf{W}_x \mathbf{x}_i + \mathbf{b}_x \in \mathbb{R}^D$, with $\mathbf{W}_x \in \mathbb{R}^{D\times T}$ and $\mathbf{b}_x \in \mathbb{R}^D$.

To represent the position within a day, we use an embedding table $\mathbf{E}_{\mathrm{tod}} \in \mathbb{R}^{M\times D}$, where M is the number of time slots in a day. Let $\boldsymbol{\tau}_t \in \mathbb{R}^D$ denote the embedding corresponding to the last time step of the input window. The history representation of entity i is defined as

$$\mathbf{s}_i = \mathrm{LN}\left(g_{\mathrm{temp}}\left([\mathbf{q}_i \| \boldsymbol{\tau}_t]\right)\right) \in \mathbb{R}^D, \tag{1}$$

where LN denotes layer normalization, $\|$ denotes concatenation, and $g_{\mathrm{temp}} : \mathbb{R}^{2D} \to \mathbb{R}^D$ is a feedforward network with two layers and a GELU activation. The representation $\mathbf{s}_i$ summarizes the input history and its position within the day and is used in the subsequent replacement.

We initialize the representation as $\mathbf{h}_i^{(0)} = \mathbf{s}_i$. At layer $\ell \in \{1, \ldots, L\}$, the representation of each entity is updated independently as $\tilde{\mathbf{h}}_i^{(\ell)} = \mathrm{LN}\Big(\mathbf{h}_i^{(\ell-1)} + g_{\mathrm{ffn}}^{(\ell)}(\mathbf{h}_i^{(\ell-1)})\Big) \in \mathbb{R}^D$, where $g_{\mathrm{ffn}}^{(\ell)} : \mathbb{R}^D \to \mathbb{R}^D$ is a feedforward network applied to each entity. Since this transformation is applied independently, interaction among entities is introduced only by the interaction defined by ERST. We collect the updated representations as $\tilde{\mathbf{H}}^{(\ell)} = [\tilde{\mathbf{h}}_1^{(\ell)}; \ldots; \tilde{\mathbf{h}}_N^{(\ell)}] \in \mathbb{R}^{N\times D}$. Figure 2 shows the forecasting process of ERST and the interaction used in each layer.

3.2 Response

Reference and Replacement. At each interaction layer, ERST defines interaction among entities from the change produced by the same function after replacing the representation of one entity with a reference. The response is defined by this comparison rather than by either representation alone.

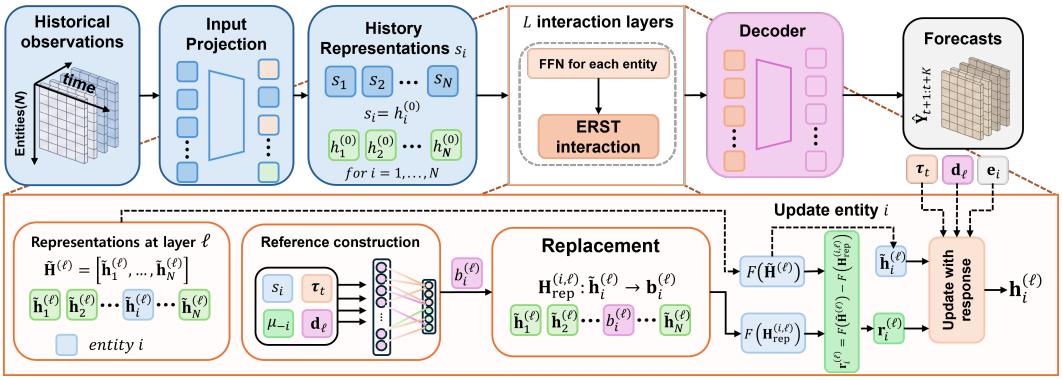


Figure 2: Overview of ERST. The upper part shows the forecasting process, while the lower part shows how the representation of one entity is replaced with a reference, how the response is obtained from the resulting change, and how the response is used to update the corresponding entity.

For entity i, the two evaluations use the same entities, normalization, and function and differ only in the representation of entity i. The resulting difference is assigned to entity i. Repeating the same comparison for each entity produces one response for each entity. The other entity representations remain unchanged in each comparison, so the response is associated with the entity whose representation is replaced.

To obtain this response, we construct a reference for each entity using its history and information from the other entities. We first map the updated representation of entity i as $\mathbf{u}_i^{(\ell)} = \phi(\tilde{\mathbf{h}}_i^{(\ell)}) \in \mathbb{R}^D$, where $\phi : \mathbb{R}^D \to \mathbb{R}^D$ is a two-layer feedforward network with layer normalization and a GELU activation, applied to each entity. The information from the other entities is then defined as

$$\boldsymbol{\mu}_{-i}^{(\ell)} = \frac{1}{N-1} \sum_{j \neq i} \mathbf{u}_j^{(\ell)} \in \mathbb{R}^D. \tag{2}$$

To construct the reference, we combine the history representation $\mathbf{s}_i$, the information from the other entities $\boldsymbol{\mu}_{-i}^{(\ell)}$, the embedding $\boldsymbol{\tau}_t$ for the position within the day, and a layer embedding $\mathbf{d}_\ell \in \mathbb{R}^D$. The reference is defined as

$$\mathbf{b}_i^{(\ell)} = \psi\left([\mathbf{s}_i \| \boldsymbol{\mu}_{-i}^{(\ell)} \| \boldsymbol{\tau}_t \| \mathbf{d}_\ell]\right) \in \mathbb{R}^D, \tag{3}$$

where $\psi : \mathbb{R}^{4D} \to \mathbb{R}^D$ is a feedforward network with two layers. The representation $\mathbf{s}_i$ provides the history of entity i, while $\boldsymbol{\mu}_{-i}^{(\ell)}$ provides information from the other entities without using the current representation of entity i. These two sources allow the reference to use information about entity i from its history and information from the other entities without directly using the current representation being replaced. Thus, $\mathbf{b}_i^{(\ell)}$ provides a point of comparison for measuring the response and is not assumed to be a uniquely determined, observed, or causal alternative.

We then replace only the representation of entity i with its reference while keeping the entity and all other representations unchanged. The representations after replacement are

$$\mathbf{H}_{\text{rep}}^{(i,\ell)} = [\tilde{\mathbf{h}}_1^{(\ell)}; \ldots; \mathbf{b}_i^{(\ell)}; \ldots; \tilde{\mathbf{h}}_N^{(\ell)}] \in \mathbb{R}^{N \times D}. \tag{4}$$

Thus, $\tilde{\mathbf{H}}^{(\ell)}$ and $\mathbf{H}_{\text{rep}}^{(i,\ell)}$ contain the same entities and differ only in the representation of entity i. We use these two representations to obtain the response for entity i.

Measuring the Response. Given $\tilde{\mathbf{H}}^{(\ell)}$, we first compute the mean of the transformed representations as $\boldsymbol{\mu}^{(\ell)} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{u}_j^{(\ell)} \in \mathbb{R}^D$. Let $\rho : \mathbb{R}^D \to \mathbb{R}^D$ denote a two-layer feedforward network with layer normalization and a GELU activation. The representation obtained from all entities is

$$F\left(\tilde{\mathbf{H}}^{(\ell)}\right) = \rho\left(\boldsymbol{\mu}^{(\ell)}\right) \in \mathbb{R}^D. \tag{5}$$

For the representations after replacement in Eq. (4), let $\mathbf{u}_{i,b}^{(\ell)} = \phi(\mathbf{b}_i^{(\ell)})$. Since only entity i is replaced, the corresponding mean is

$$\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)} = \boldsymbol{\mu}^{(\ell)} + \frac{\mathbf{u}_{i,b}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N}, \tag{6}$$

Only the contribution of entity i changes, while the contributions of all other entities remain unchanged. Thus, the two sets of representations differ only in the representation of entity i. The entities, their number, the normalization, and the function remain unchanged. This separates the change produced by replacing one representation from changes caused by removing an entity or changing the normalization. In contrast, removing entity i changes both the set of entities and the normalization. A detailed comparison is provided in Appendix B.

We define the response for entity i as

$$\mathbf{r}_i^{(\ell)} = F\Big(\tilde{\mathbf{H}}^{(\ell)}\Big) - F\Big(\mathbf{H}_{\mathrm{rep}}^{(i,\ell)}\Big) = \rho\Big(\boldsymbol{\mu}^{(\ell)}\Big) - \rho\Big(\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)}\Big) \in \mathbb{R}^D. \tag{7}$$

The same function ρ is evaluated before and after replacing the representation of entity i. Thus, $\mathbf{r}_i^{(\ell)}$ is the difference produced by this replacement. We write this difference as

$$\mathcal{D}_i^{\mathfrak{b}} F\Big(\tilde{\mathbf{H}}^{(\ell)}\Big) = F\Big(\tilde{\mathbf{H}}^{(\ell)}\Big) - F\Big(\mathbf{H}_{\mathrm{rep}}^{(i,\ell)}\Big) = \mathbf{r}_i^{(\ell)}. \tag{8}$$

For a given reference, ERST defines the interaction for entity i as the response obtained from the same function before and after replacing only its representation. The reference specifies the point of comparison, whereas the resulting difference defines the response used as interaction information for entity i. If $\mathbf{u}_{i,b}^{(\ell)} = \mathbf{u}_i^{(\ell)}$, then $\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)} = \boldsymbol{\mu}^{(\ell)}$ and $\mathbf{r}_i^{(\ell)} = \mathbf{0}$.

Let the change in the transformed representation of entity i be

$$\Delta\mathbf{u}_i^{(\ell)} = \mathbf{u}_{i,b}^{(\ell)} - \mathbf{u}_i^{(\ell)}. \tag{9}$$

Under a local smoothness assumption on ρ, the response can be approximated to first order as

$$\mathbf{r}_i^{(\ell)} \approx -\frac{1}{N}\mathbf{J}_\rho\Big(\boldsymbol{\mu}^{(\ell)}\Big)\,\Delta\mathbf{u}_i^{(\ell)}, \tag{10}$$

where $\mathbf{J}_\rho$ denotes the Jacobian of ρ. The response depends on both the change in the representation of entity i and the function evaluated at the current mean. Thus, the same change in one entity can produce different responses when the representations of the other entities differ.

The representations after replacement need not be reconstructed explicitly for every entity. Once $\boldsymbol{\mu}^{(\ell)}$ is computed, the mean after each replacement is obtained by changing only the contribution of one entity. The approximation, the distinction from removing an entity, and the permutation equivariance of the response are provided in Appendix B; the complexity analysis is given in Appendix D.

3.3 Using the Response

The response in Eq. (7) provides interaction information for each entity. The following operations determine how this response is incorporated into the forecasting representation and do not change the definition of the response. We incorporate the response using information about the entity and time. Each entity is assigned an embedding $\mathbf{e}_i \in \mathbb{R}^D$. This embedding is not used to construct the reference and is used only when the response is incorporated into the representation.

As shown in Figure 3, the response is used together with information about the entity, time, and layer. At layer ℓ, we construct

$$\mathbf{g}_i^{(\ell)} = \sigma(\mathbf{W}_g\,\mathrm{LN}\,(\mathbf{e}_i + \boldsymbol{\tau}_t + \mathbf{d}_\ell) + \mathbf{b}_g) \in (0,1)^D, \tag{11}$$

where σ denotes the sigmoid function, $\mathbf{W}_g \in \mathbb{R}^{D\times D}$, and $\mathbf{b}_g \in \mathbb{R}^D$. The response is then written as

$$\bar{\mathbf{r}}_i^{(\ell)} = g_r\Big(\mathbf{r}_i^{(\ell)}\Big) \odot \mathbf{g}_i^{(\ell)} \in \mathbb{R}^D, \tag{12}$$

where $g_r : \mathbb{R}^D \to \mathbb{R}^D$ is a learnable projection with a GELU activation and $\odot$ denotes element-wise multiplication. The vector $\bar{\mathbf{r}}_i^{(\ell)}$ is the response used in the update.

The response is combined with the updated representation of entity i as $\mathbf{z}_i^{(\ell)} = [\tilde{\mathbf{h}}_i^{(\ell)} \| \bar{\mathbf{r}}_i^{(\ell)}] \in \mathbb{R}^{2D}$. From $\mathbf{z}_i^{(\ell)}$, we compute $\mathbf{v}_i^{(\ell)} = g_u(\mathbf{z}_i^{(\ell)}) \in \mathbb{R}^D$, where $g_u : \mathbb{R}^{2D} \to \mathbb{R}^D$ is a feedforward network with two layers. We also compute

$$\boldsymbol{\alpha}_i^{(\ell)} = \sigma\left(\mathbf{W}_\alpha \mathbf{z}_i^{(\ell)} + \mathbf{b}_\alpha\right) \in (0,1)^D, \tag{13}$$

where $\mathbf{W}_\alpha \in \mathbb{R}^{D \times 2D}$ and $\mathbf{b}_\alpha \in \mathbb{R}^D$. This gate determines how much $\mathbf{v}_i^{(\ell)}$ contributes to the update. The representation at layer ℓ is then

$$\mathbf{h}_i^{(\ell)} = \mathrm{LN}\left(\tilde{\mathbf{h}}_i^{(\ell)} + \boldsymbol{\alpha}_i^{(\ell)} \odot \mathbf{v}_i^{(\ell)}\right) \in \mathbb{R}^D. \tag{14}$$

The representation $\tilde{\mathbf{h}}_i^{(\ell)}$ is retained in the update, while the response provides the interaction information. Additional analysis is provided in Appendix C.

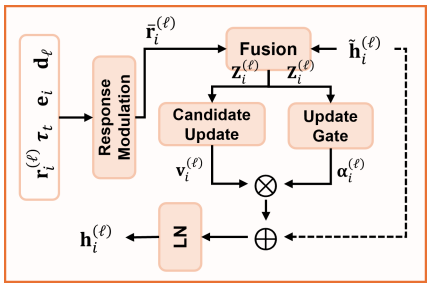


Figure 3: Update for entity i at layer ℓ. The response is modulated using information about the entity, time, and layer and combined with the entity representation for the update.

After L interaction layers, a decoder $g_{\mathrm{dec}} : \mathbb{R}^D \to \mathbb{R}^K$ maps the final representation to $\boldsymbol{\delta}_i = g_{\mathrm{dec}}(\mathbf{h}_i^{(L)})$. The final prediction is $\hat{\mathbf{y}}_i = \bar{x}_{t,i} + \boldsymbol{\delta}_i$, where the last projected input value $\bar{x}_{t,i}$ is used across the K forecasting steps. The same procedure is applied to all entities.

4 Experiments

Our experiments examine three questions: **(Q1)** whether the response provides information beyond the representation itself, **(Q2)** whether the response depends on replacing one entity representation while keeping the others unchanged and using it for the corresponding entity, and **(Q3)** whether the same interaction can be used across different forecasting architectures and system scales.

4.1 Setup

Datasets and Protocols. We evaluate ERST on spatiotemporal forecasting datasets from traffic sensing, urban mobility, air quality, wind power, and mobile communication systems. The evaluation includes PEMS03, PEMS04, PEMS07, and PEMS08 (Guo et al., 2022), KnowAir (Wu et al., 2026; Wang et al., 2020), SDWPF (Zhou et al., 2024), Milan Internet (Barlacchi et al., 2015; Zhang et al., 2019), NYCTaxi, CHIBike, and T-Drive (Jiang et al., 2023a), and the CA and GLA subsets of LargeST (Liu et al., 2023; Fang et al., 2025).

For PeMS, KnowAir, SDWPF, Milan Internet, and LargeST, we use the previous 12 observations to predict the next 12 steps with chronological training, validation, and test splits of 6:2:2. For NYCTaxi, CHIBike, and T-Drive, we follow the grid forecasting protocol of PDFormer (Jiang et al., 2023a), using the previous six observations to predict the next step with a 7:1:2 split. Normalization statistics are computed only from the training set. Detailed dataset statistics, forecasting protocols, baseline descriptions, and implementation settings are provided in Appendix E.

Baselines. We compare ERST with classical methods, methods using graphs, methods using Transformers, methods without graphs, and recent spatiotemporal forecasting methods (Li et al., 2018; Wu et al., 2019; Shao et al., 2022; Nie et al., 2023; Liu et al., 2024; Han et al., 2024b; Fang et al., 2025; Zhao et al., 2026). The baseline methods for each dataset follow the corresponding evaluation protocol. Complete forecasting results are provided in Appendix F.

Implementation Details. By default, ERST uses a representation dimension of $D = 64$, $L = 2$ interaction layers, and a dropout rate of 0.1. We use a batch size of 32 and optimize the model with AdamW using an initial learning rate of 10^{-3} and weight decay of 10^{-4}. The maximum number of training epochs is 100, and early stopping is applied based on validation MAE with a patience of 15 epochs. The model achieving the lowest validation MAE is used for test evaluation. Additional analyses of the number of interaction layers and the proportion of participating entities are provided in Appendix H. Code: `https://anonymous.4open.science/r/ERST-C2F4/`

Table 1: Forecasting performance on representative spatiotemporal datasets. Lower values are better. Best results are shown in bold and second-best results are underlined.

	PEMS03		PEMS07		KnowAir		SDWPF		NYCTaxi-In		T-Drive-In	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
STGCN	17.05	29.42	24.59	38.78	16.66	25.40	119.11	189.50	14.37	23.87	22.12	38.65
GWNet	14.61	25.23	21.22	34.11	14.89	22.18	96.88	165.85	14.31	23.79	19.55	36.15
AGCRN	16.63	28.36	21.13	34.69	15.53	23.83	100.39	169.54	14.93	25.50	24.77	68.22
MTGNN	14.91	25.43	21.02	33.89	14.85	23.36	94.19	165.75	14.17	23.61	20.41	36.93
STID	15.28	26.85	21.61	32.79	15.04	23.71	102.78	178.09	14.14	23.59	17.90	31.21
PatchTST	21.04	33.45	30.36	47.13	15.79	24.95	96.72	171.20	16.32	30.96	29.26	64.50
iTransformer	18.02	28.79	26.40	41.67	15.31	24.31	95.12	166.37	19.88	32.93	33.13	63.17
SOFTS	18.01	28.64	26.15	41.28	15.34	24.37	97.14	167.15	20.39	33.49	24.89	55.19
TWIST	15.11	24.76	19.70	32.70	19.92	29.16	140.47	208.34	13.65	23.00	17.96	31.90
RAST	15.89	25.34	21.28	34.21	15.19	23.83	98.56	174.30	14.64	24.63	22.95	44.33
FaST	15.41	25.10	21.22	36.11	14.84	23.74	95.32	168.14	15.07	25.10	20.45	37.56
ERST (Ours)	**14.28**	**22.20**	**19.51**	**31.96**	**13.98**	**20.50**	**93.08**	**131.20**	**13.43**	**22.69**	**16.56**	**29.05**

Table 2: Forecasting performance on the LargeST benchmarks. Lower values are better. Best results are shown in bold and second-best results are underlined.

Dataset	Metric	STID	PatchTST	SOFTS	TWIST	PatchSTG	RAST	FaST	**ERST (Ours)**
CA	MAE	18.61	46.32	33.12	24.88	**17.65**	29.30	23.95	18.03
	RMSE	32.00	75.65	56.29	36.73	30.25	49.13	44.19	**29.67**
	MAPE	14.25	32.37	24.66	18.43	12.85	21.71	17.25	**12.13**
GLA	MAE	21.03	51.07	34.72	34.88	**19.32**	31.65	24.87	19.80
	RMSE	34.68	82.65	59.16	50.20	32.63	52.20	43.40	**32.13**
	MAPE	14.46	30.54	22.17	28.95	11.48	20.76	15.33	**11.23**

Evaluation Metrics. We evaluate forecasting performance using mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE), where lower values indicate better performance.

4.2 Overall Forecasting Performance

Table 1 compares ERST with eleven forecasting methods, including methods using graphs, methods using Transformers, and methods without graphs, across six sensor and grid datasets. ERST achieves the lowest MAE and RMSE on all six datasets. The results are observed across traffic, air quality, wind power, and urban mobility datasets, showing that the same interaction can be used across different spatiotemporal domains. For LargeST-CA and LargeST-GLA, which contain larger numbers of entities, Table 2 shows that ERST achieves the second lowest MAE and the lowest RMSE on both datasets. The results also show that the interaction can be used when the number of entities increases. Complete results and evaluations over longer forecasting horizons are provided in Appendix F and Appendix G, respectively.

4.3 Ablation Study

Response and Reference. To address Q1, we first examine whether the response provides information beyond the representation before or after replacement. *Original Representation* uses the representation obtained before replacement, *Replaced Representation* uses the representation obtained after replacement, and *w/o Response* removes the response from the interaction. These variants test whether the difference produced by the replacement is needed as interaction information.

We also examine the constraints used to construct the reference using $\psi = 0$, *w/o History*, and *w/o Other Entities*. These variants respectively set the reference to zero, remove the history representation $\mathbf{s}_i$, and remove information from the other entities $\boldsymbol{\mu}_{-i}$. They test the role of the reference and whether its construction benefits from retaining the history of entity i and information from the other entities without directly using the current representation being replaced.

Replacement and Use of the Response. To address Q2, *Remove Entity* replaces the proposed operation with removal of entity i. This changes both the set of entities and the normalization, whereas ERST keeps them unchanged and replaces only the representation of entity i. *w/o Modulation and Gating* keeps the response but removes the corresponding operations used when incorporating it into the representation. This variant examines whether using the response in the update contributes beyond constructing the response itself. Table 3 reports the results.

Table 3: Ablation results on PEMS03, SDWPF, and T-Drive. Lower values are better. Best results are shown in bold.

	PEMS03		SDWPF		T-Drive-In	
Variant	MAE	RMSE	MAE	RMSE	MAE	RMSE
Original Representation	15.68	23.69	95.31	133.57	22.95	39.33
Replaced Representation	15.65	23.68	94.65	133.56	22.74	39.31
w/o Response	16.23	24.63	98.98	139.75	24.01	41.91
$\psi = 0$	14.38	22.62	95.00	133.71	16.80	29.89
w/o History	14.58	22.88	95.12	133.74	17.81	31.30
w/o Other Entities	14.52	22.60	96.33	135.52	18.04	31.93
Remove Entity	14.33	22.69	94.90	135.79	18.15	31.22
w/o Modulation and Gating	15.67	23.77	95.39	135.21	22.66	39.60
ERST	**14.28**	**22.20**	**93.08**	**131.20**	**16.56**	**29.05**

ERST achieves the lowest MAE and RMSE on all three datasets. For Q1, using either the original or replaced representation, or removing the response, degrades performance across the three datasets. Even with $\psi = 0$, using the response performs better than removing the response, showing that the improvement does not depend only on the learned reference. Using the history of the entity and information from the other entities further reduces forecasting error when constructing the reference.

For Q2, *Remove Entity* performs worse than ERST, and the difference remains statistically significant across all seven evaluated datasets in experiments with different random seeds. Removing modulation and gating also degrades performance. These results distinguish replacing one representation while keeping the entities and normalization unchanged from removing an entity, and show that using the response in the update contributes to forecasting performance. Extended results are provided in Appendix H.

4.4 Generalization across Forecasting Architectures

To address Q3, we examine whether the interaction defined by ERST can be used with different forecasting architectures. We evaluate ERST with four architectures—LSTM, STID, iTransformer, and PatchTST—using the same definition of the response and the same procedure for using it. Figure 4 reports MAE on six datasets and compares each original architecture with its corresponding ERST variant.

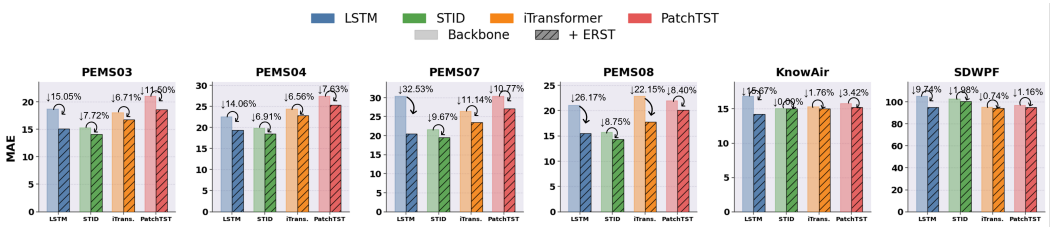


Figure 4: Results of ERST with different forecasting architectures. Each panel compares the MAE of the original architecture and its corresponding ERST variant on one dataset. ERST is evaluated with LSTM, STID, iTransformer, and PatchTST. Lower values are better.

ERST improves forecasting accuracy across nearly all evaluated combinations of forecasting architecture and dataset. For Q3, the improvement is observed across different forecasting architectures

and across datasets from traffic, air quality, and wind power. In all cases, ERST constructs interaction from the response obtained before and after replacing one entity representation, without changing how interaction is defined for each architecture. These results show that the response can provide interaction information across different forecasting architectures and spatiotemporal domains. Additional results across forecasting architectures are provided in Appendix I.

4.5 Response and Entity

We examine whether each response is associated with the entity for which it was computed. At inference time, we apply *Response Shuffle*, which reassigns response vectors across entities while preserving their values. Only the assignment of responses to entities is changed.

As shown in Figure 5, Response Shuffle increases MAE on all five datasets. Since the response values are preserved while their assignments are changed, the result shows that the responses are not interchangeable across entities. This provides evidence that each response contains information associated with the entity for which it was computed. Additional analyses of the response are provided in Appendix I.

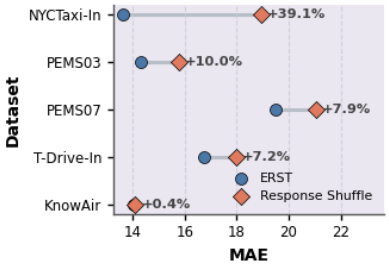


Figure 5: Effect of reassigning responses across entities. Percentages denote the relative MAE increase over ERST.

4.6 Efficiency

We examine the computational cost of ERST for systems with different numbers of entities. Table 4 reports the number of parameters, training time, and inference time on PEMS07, SDWPF, and LargeST-CA. In the response computation, each mean after replacement is obtained from the mean before replacement by changing only the contribution of one entity. Thus, ERST does not reconstruct all entity representations for every replacement or construct dense relations between pairs of entities. For a fixed representation dimension, the computation of the responses is linear in the number of entities. Additional analyses of runtime and the number of entities are provided in Appendix J and Appendix D.

Table 4: Efficiency comparison on the evaluated datasets. Par. denotes the number of trainable parameters in millions (M). Train denotes the average training time per epoch (s/epoch), and Infer denotes the total inference time (s). Lower values are better. The lowest values are shown in bold and the second lowest values are underlined.

	PEMS07				SDWPF				LargeST-CA			
Model	MAE	Par.	Train	Infer	MAE	Par.	Train	Infer	MAE	Par.	Train	Infer
STID	21.61	**0.140**	**7.55**	**1.17**	102.78	**0.116**	**4.22**	**0.76**	18.61	1.511	<u>94.96</u>	61.96
TWIST	<u>19.70</u>	1.121	42.22	<u>3.18</u>	140.47	0.719	<u>14.80</u>	<u>1.14</u>	24.88	5.072	507.88	35.93
PatchSTG	20.37	2.302	22.25	4.37	98.66	2.249	38.17	3.60	**17.65**	1.841	488.70	48.73
FaST	21.22	0.485	32.14	3.68	<u>95.32</u>	0.399	18.01	2.04	23.95	<u>1.226</u>	155.71	<u>14.98</u>
ERST (Ours)	**19.51**	<u>0.256</u>	<u>14.55</u>	<u>1.52</u>	**93.08**	<u>0.199</u>	15.66	1.35	<u>18.03</u>	**0.750**	**79.81**	**12.90**

5 Conclusion

We introduced ERST for spatiotemporal forecasting. ERST defines interaction among entities from the change produced by the same function after replacing the representation of one entity with a reference while keeping the entities, normalization, and function unchanged. The resulting response is used to update the representation of that entity and provides interaction information beyond the representation itself. Evaluations across multiple domains, system scales, and forecasting architectures show that the same interaction can be used in different spatiotemporal forecasting settings.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, literature retrieval and discovery, research brainstorming, code development, feedback on experimental design, and the drafting, organization, and presentation of the manuscript and experimental results. The final methodology, experiments, mathematical claims, and scientific conclusions were determined and validated by the authors. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021. doi: 10.1016/j.artint.2021.103502.

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

Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific Data*, 2: 150055, 2015. doi: 10.1038/sdata.2015.55.

Lingxiao Cao, Bin Wang, Guiyuan Jiang, Yanwei Yu, and Junyu Dong. Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11463–11471, 2025.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, volume 9, pp. 155–161, 1996.

Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 307–317, 2025. doi: 10.1145/3690624.3709177.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 364–373, 2021.

Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437, 2017.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. In *International Conference on Learning Representations*, 2021.

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

Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proceedings of the VLDB Endowment*, 17(5):1081–1090, 2024a.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-2046.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4365–4373, 2023a. doi: 10.1609/aaai.v37i4.25556.

Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8078–8086, 2023b.

Weiyang Kong, Ziyu Guo, and Yubao Liu. Spatio-temporal pivotal graph neural networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8): 8627–8635, Mar. 2024. doi: 10.1609/aaai.v38i8.28707. URL https://ojs.aaai.org/index.php/AAAI/article/view/28707.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3744–3753, 2019.

Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4189–4196, 2021. doi: 10.1609/aaai.v35i5.16542.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

Yujie Li, Zezhi Shao, Yongjun Xu, Qiang Qiu, Zhaogang Cao, and Fei Wang. Dynamic frequency domain graph convolutional network for traffic forecasting. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5245–5249, 2024. doi: 10.1109/ICASSP48485.2024.10446144.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Bei Pan, Ugur Demiryurek, and Cyrus Shahabi. Utilizing real-world transportation data for accurate traffic prediction. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pp. 595–604, USA, 2012. IEEE Computer Society. ISBN 9780769549057. doi: 10.1109/ICDM.2012.52. URL https://doi.org/10.1109/ICDM.2012.52.

Weilin Ruan, Xilin Dang, Ziyu Zhou, Sisuo Lyu, and Yuxuan Liang. A retrieval augmented spatio-temporal framework for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 39163–39172, 2026. doi: 10.1609/aaai.v40i46.41264.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information Knowledge Management*, pp. 4454–4458, 2022.

Jayaraman J. Thiagarajan, Vivek Sivaraman Narayanaswamy, Deepta Rajan, Jason Liang, Akshay Chaudhari, and Andreas Spanias. Designing counterfactual generators using deep model inversion. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16873–16884, 2021.

Peng Wang, Longxi Feng, Wenhao Zhang, and Kanghua Hui. Twist: An efficient spatial—temporal transformer with temporal window and sparse attention for traffic forecasting. *IEEE Internet of Things Journal*, 12(14):26799–26815, 2025. doi: 10.1109/JIOT.2025.3561542.

Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2.5-gnn: A domain knowledge enhanced graph neural network for pm2.5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pp. 163–166, 2020. doi: 10.1145/3397536.3422208.

Binqing Wu, Zongjiang Shang, Shiyu Liu, Jianlong Huang, Jiahui Xu, and Ling Chen. Airdde: multifactor neural delay differential equations for air quality forecasting. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i19.38627. URL https://doi.org/10.1609/aaai.v40i19.38627.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, 2019. doi: 10.24963/ijcai.2019/264.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 37068–37088. Curran Associates, Inc., 2023. doi: 10.52202/075280-1611. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/74fa3651b41560e1c7555e0958c70333-Paper-Conference.pdf.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity of explanations. In *Advances in Neural Information Processing Systems*, volume 32, pp. 10965–10976, 2019.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/505. URL https://doi.org/10.24963/ijcai.2018/505.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

Chuanting Zhang, Haixia Zhang, Jingping Qiao, Dongfeng Yuan, and Minggao Zhang. Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data. *IEEE Journal on Selected Areas in Communications*, 37(6):1389–1401, 2019. doi: 10.1109/JSAC.2019.2904363.

Qianru Zhang, Xinyi Gao, Haixin Wang, Siu Ming Yiu, and Hongzhi Yin. Efficient traffic prediction through spatio-temporal distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1093–1101, 2025.

Yiji Zhao, Zihao Zhong, Ao Wang, Haomin Wen, Ming Jin, Yuxuan Liang, Huaiyu Wan, and Hao Wu. Fast: Efficient and effective long-horizon forecasting for large-scale spatial-temporal graphs via mixture-of-experts. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '26, pp. 1975–1986, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722585. doi: 10.1145/3770854.3780165. URL https://doi.org/10.1145/3770854.3780165.

Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jian Tang, Jiantao Su, Yu Li, Ji Liu, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting over a large turbine array. *Scientific Data*, 11:649, 2024. doi: 10.1038/s41597-024-03427-5.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022.

A Empirical Motivation

Empirical Question. We begin by examining how observed variation is distributed across entities in spatiotemporal systems. If observed changes are distributed similarly across entities, a representation formed from all entities can summarize broadly shared variation. In contrast, if the observed variation is concentrated in a subset of entities, it is useful to examine not only the representation formed from all entities, but also how that representation changes with respect to one entity.

To examine this property independently of ERST, we use only observations from the training portion of each dataset. No trained ERST model, learned representation, or forecasting output is used in this analysis. The purpose of this analysis is not to evaluate ERST itself, but to examine how observed changes are distributed across entities and to motivate measuring the change associated with one entity.

Let $x_{t,i}$ denote the scalar forecasting target of entity i at time t. For entity i at time t, we define a recent reference value using the mean of the preceding three observations,

$$\bar{x}_{t,i}^{\text{recent}} = \frac{x_{t-1,i} + x_{t-2,i} + x_{t-3,i}}{3}. \tag{15}$$

To account for differences in scale across entities, we define the standardized state-change magnitude as

$$d_{t,i} = \frac{\left| x_{t,i} - \bar{x}_{t,i}^{\text{recent}} \right|}{\sigma_i}, \tag{16}$$

where σ_i is the standard deviation of entity i computed from the training portion of the corresponding dataset.

At each time step, we convert the proportion p into an integer number of entities,

$$k_p = \text{round}\left(\frac{pN}{100}\right), \tag{17}$$

and let $\mathcal{S}_p(t)$ denote the set containing the k_p entities with the largest values of $d_{t,i}$. We measure the fraction of the total standardized change accounted for by these entities as

$$C_p(t) = \frac{\sum_{i \in \mathcal{S}_p(t)} d_{t,i}}{\sum_{j=1}^{N} d_{t,j}}. \tag{18}$$

We report the average of this fraction over the training time steps with nonzero total standardized change as

$$C_p = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} C_p(t), \tag{19}$$

where $\mathcal{T}$ denotes the training time steps for which $\sum_{j=1}^{N} d_{t,j} > 0$.

Table 5: Distribution of observed changes across entities using training observations only. Top 10%, Top 20%, and Top 30% Share denote the percentage of the total standardized change accounted for by the selected k_p entities with the largest values of $d_{t,i}$, where k_p is the rounded number of entities corresponding to $p\%$ of all entities. Gain@10 compares the Top-10% share with the selected fraction k_{10}/N.

Dataset	Top 10% Share	Top 20% Share	Top 30% Share	Gain@10
PEMS03	30.21	48.33	61.82	3.00
PEMS04	33.50	51.63	65.39	3.32
PEMS07	32.95	51.65	65.24	3.31
PEMS08	32.50	51.02	64.71	3.25
KnowAir	32.10	51.25	64.56	3.28
SDWPF	35.31	53.61	66.03	3.64
Milan Internet	26.75	44.05	57.86	2.68
NYCTaxi-In	34.97	51.56	64.17	3.28
CHIBike-In	80.52	97.94	100.00	8.05
T-Drive-In	30.69	48.28	61.35	3.08
LargeST-CA	30.67	48.29	61.76	3.07
LargeST-GLA	30.09	47.64	61.09	3.01

We additionally report

$$G_p = \frac{C_p}{k_p/N}, \tag{20}$$

which compares the share of the total standardized change with the fraction of entities selected after integer rounding. A value of $G_p > 1$ indicates that the selected entities account for a larger share of the total change than their fraction of all entities. For example, if approximately 10% of the entities account for approximately 30% of the total standardized change, then G_p is approximately 3.

Observed Distribution of Changes. Table 5 shows that the entities with the largest values of $d_{t,i}$ account for a larger share of the total standardized change than their fraction of all entities. Excluding CHIBike-In, the Top-10% entities account for 26.75%–35.31% of the total standardized change, with Gain@10 values of 2.68–3.64. The Top-20% entities account for 44.05%–53.61% of the total change, while the Top-30% entities account for 57.86%–66.03%. For CHIBike-In, the Top-10% entities account for 80.52% of the total standardized change. This concentration is observed across traffic sensing, urban mobility, air quality, wind power, mobile communication, and the traffic datasets in LargeST.

From the Observation to the Response. The results above show that observed changes are not distributed uniformly across entities in the evaluated systems. This motivates examining not only the representation formed from all entities, but also how that representation changes with respect to one entity:

> How does the representation formed from all entities change when the representation of one entity is replaced?

A representation $F(\mathbf{H})$ describes the entities jointly, but does not directly describe how that representation changes with respect to one entity. To examine this change, we compare two sets of representations that contain the same entities and differ only in the representation of entity i.

Removing entity i would change the set of entities and the normalization. ERST instead keeps the entity and replaces only its representation with a reference $\mathbf{b}_i$:

$$\mathbf{H} = [\mathbf{h}_1; \ldots; \mathbf{h}_i; \ldots; \mathbf{h}_N] \quad \longrightarrow \quad \mathbf{H}^{(i)}_{\text{rep}} = [\mathbf{h}_1; \ldots; \mathbf{b}_i; \ldots; \mathbf{h}_N]. \tag{21}$$

The two sets therefore contain the same entities and differ only in the representation of entity i.

ERST defines the response for entity i as

$$\mathbf{r}_i = F(\mathbf{H}) - F(\mathbf{H}^{(i)}_{\text{rep}}). \tag{22}$$

The empirical analysis motivates examining the change associated with one entity, while ERST defines this change as the response and uses it as interaction information for forecasting.

Relation Between the Empirical Analysis and ERST. The Top-10%, Top-20%, and Top-30% analyses above are not part of ERST. They use only the observed data to examine how changes are distributed across entities. ERST does not select only the entities with the largest changes and does not replace multiple entities at the same time. For each entity i, ERST constructs a reference, replaces only its representation, and measures the resulting change using the same function.

The empirical analysis therefore does not directly evaluate ERST. It provides an observation that motivates examining how a representation formed from all entities changes with respect to one entity. ERST addresses this question through the following sequence:

representation formed from all entities
↓
replace the representation of one entity
↓
apply the same function
↓
use the resulting difference as the response.

How the response is incorporated into the representation is analyzed separately in Appendix C.

B Theoretical Interpretation of the Response

This section provides a mathematical interpretation of the response for a given reference. The reference specifies the representation used for comparison and is not interpreted as a physical or causal alternative. We analyze the response as the difference obtained before and after replacing the representation of one entity, together with the corresponding change in its transformed representation, the sensitivity of the same function, and the fact that the entities, their number, and normalization remain unchanged during the comparison.

B.1 Response under Replacement

At interaction layer ℓ, the mean before replacement is

$$\boldsymbol{\mu}^{(\ell)} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{u}_j^{(\ell)}, \tag{23}$$

and the mean after replacing the representation of entity i is

$$\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)} = \boldsymbol{\mu}^{(\ell)} + \frac{\mathbf{u}_{i,b}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N}, \tag{24}$$

where $\mathbf{u}_{i,b}^{(\ell)} = \phi(\mathbf{b}_i^{(\ell)})$. The two means are computed from the same entities and use the same normalization. They differ only in the contribution of entity i, whose representation is replaced by the reference.

For compactness, define the change in the transformed representation of entity i as

$$\Delta\mathbf{u}_i^{(\ell)} = \mathbf{u}_{i,b}^{(\ell)} - \mathbf{u}_i^{(\ell)}. \tag{25}$$

Then

$$\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)} = \boldsymbol{\mu}^{(\ell)} + \frac{1}{N}\Delta\mathbf{u}_i^{(\ell)}. \tag{26}$$

Using the same function ρ, the response is

$$\mathbf{r}_i^{(\ell)} = \rho\Big(\boldsymbol{\mu}^{(\ell)}\Big) - \rho\Big(\boldsymbol{\mu}^{(\ell)} + \frac{1}{N}\Delta\mathbf{u}_i^{(\ell)}\Big). \tag{27}$$

Equation (27) shows that the response is the difference obtained by applying the same function before and after replacing the representation of entity i. It therefore uses the change produced by the replacement rather than either representation alone.

B.2 Role of the Reference

The reference provides the point of comparison used to obtain the response. It is constructed without directly using the current representation being replaced. The history representation $\mathbf{s}_i$ provides information about entity i from its observations, while $\boldsymbol{\mu}_{-i}^{(\ell)}$ provides information from the other entities without including the current representation of entity i. Together with the time and layer embeddings, these quantities define $\mathbf{b}_i^{(\ell)}$ without directly reusing $\tilde{\mathbf{h}}_i^{(\ell)}$.

The purpose of this construction is to retain information about the entity and the other entities while changing the representation whose effect is being measured. The reference specifies the point of comparison, whereas the interaction is defined by the resulting difference between the two evaluations of the same function. The reference is not assumed to be unique or optimal, nor is it interpreted as an observed or causal alternative. Instead, its role is to provide a representation for comparison without directly reusing the representation being replaced.

The ablations in Appendix H examine the information used to construct the reference. Setting the reference to zero, removing the history of the entity, or removing information from the other entities increases forecasting error. These results support using these sources of information to construct the reference used for comparison.

B.3 Local Approximation

The following result provides a local approximation of the response and a bound on the approximation error.

Proposition (Local approximation of the response). Let

$$\boldsymbol{\delta}_i^{(\ell)} = \frac{1}{N}\Delta\mathbf{u}_i^{(\ell)}. \tag{28}$$

Assume that the Jacobian of ρ is L_ρ-Lipschitz under the induced operator norm on the line segment connecting $\boldsymbol{\mu}^{(\ell)}$ and $\boldsymbol{\mu}^{(\ell)} + \boldsymbol{\delta}_i^{(\ell)}$. Then

$$\mathbf{r}_i^{(\ell)} = -\frac{1}{N}\mathbf{J}_\rho\Big(\boldsymbol{\mu}^{(\ell)}\Big)\,\Delta\mathbf{u}_i^{(\ell)} + \boldsymbol{\epsilon}_i^{(\ell)}, \tag{29}$$

where

$$\left\|\boldsymbol{\epsilon}_i^{(\ell)}\right\| \le \frac{L_\rho}{2N^2}\left\|\Delta\mathbf{u}_i^{(\ell)}\right\|^2. \tag{30}$$

Proof. By the first-order Taylor theorem with a Lipschitz Jacobian,

$$\rho\left(\boldsymbol{\mu} + \boldsymbol{\delta}\right) = \rho\left(\boldsymbol{\mu}\right) + \mathbf{J}_\rho\left(\boldsymbol{\mu}\right)\boldsymbol{\delta} + \boldsymbol{\eta}, \tag{31}$$

where

$$\|\boldsymbol{\eta}\| \le \frac{L_\rho}{2}\|\boldsymbol{\delta}\|^2. \tag{32}$$

Setting

$$\boldsymbol{\delta} = \frac{1}{N}\Delta\mathbf{u}_i^{(\ell)} \tag{33}$$

and substituting it into Eq. (27) gives Eq. (29). With $\boldsymbol{\epsilon}_i^{(\ell)} = -\boldsymbol{\eta}$, the bound in Eq. (30) follows.

Equation (29) shows that the response depends on two terms. The change $\Delta\mathbf{u}_i^{(\ell)}$ describes how the transformed representation of entity i changes after replacement, while $\mathbf{J}_\rho(\boldsymbol{\mu}^{(\ell)})$ describes the sensitivity of the function at the current mean. Therefore, the same change in the representation of entity i can produce different responses when the representations of the other entities differ, because the local behavior of ρ can also differ.

Locally, the response can be summarized as

$$\begin{aligned}\text{change in the transformed representation} \;&\times\; \text{sensitivity of the function}\\ &\longrightarrow\; \text{response.}\end{aligned} \tag{34}$$

The multiplication symbol denotes the Jacobian–vector operation in Eq. (29) and is not interpreted as a causal effect.

B.4 Response Bound

Replacing the representation of one entity changes the mean by $\Delta\mathbf{u}_i^{(\ell)}/N$. The following result bounds the magnitude of the corresponding response.

Corollary (Response bound). If ρ is K_ρ-Lipschitz on the line segment connecting $\boldsymbol{\mu}^{(\ell)}$ and $\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)}$, then

$$\left\|\mathbf{r}_i^{(\ell)}\right\| \leq \frac{K_\rho}{N}\left\|\Delta\mathbf{u}_i^{(\ell)}\right\|. \tag{35}$$

Proof. Applying Lipschitz continuity to Eq. (27) gives

$$\left\|\mathbf{r}_i^{(\ell)}\right\| \leq K_\rho\left\|\frac{1}{N}\Delta\mathbf{u}_i^{(\ell)}\right\|, \tag{36}$$

which gives Eq. (35).

The bound shows that the factor $1/N$ comes from the contribution of one entity to the mean and is not introduced by an additional normalization in the forecasting process. For interpretation, Eq. (29) can also be written as

$$N\mathbf{r}_i^{(\ell)} = -\mathbf{J}_\rho\left(\boldsymbol{\mu}^{(\ell)}\right)\Delta\mathbf{u}_i^{(\ell)} + N\boldsymbol{\epsilon}_i^{(\ell)}, \tag{37}$$

with

$$\left\|N\boldsymbol{\epsilon}_i^{(\ell)}\right\| \leq \frac{L_\rho}{2N}\left\|\Delta\mathbf{u}_i^{(\ell)}\right\|^2. \tag{38}$$

The scaled response $N\mathbf{r}_i^{(\ell)}$ is used only for this interpretation. ERST uses $\mathbf{r}_i^{(\ell)}$ directly in forecasting.

B.5 Replacement versus Entity Removal

ERST keeps entity i and replaces only its representation. The difference between replacement and removing an entity can be seen directly from the corresponding means.

Replacement. When the representation of entity i is replaced,

$$\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)} - \boldsymbol{\mu}^{(\ell)} = \frac{\mathbf{u}_{i,b}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N}. \tag{39}$$

The contributions of all other entities remain unchanged, and the number of entities remains N.

Removal. If entity i is removed, the mean becomes

$$\boldsymbol{\mu}_{-i}^{(\ell)} = \frac{1}{N-1}\sum_{j\neq i}\mathbf{u}_j^{(\ell)} = \frac{N\boldsymbol{\mu}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N-1}, \tag{40}$$

and therefore

$$\boldsymbol{\mu}_{-i}^{(\ell)} - \boldsymbol{\mu}^{(\ell)} = \frac{\boldsymbol{\mu}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N-1}. \tag{41}$$

The two operations therefore differ in both the representation being changed and the normalization used to compute the mean. ERST keeps the same entities and normalization and changes only the representation of entity i, whereas removal changes the set of entities and replaces the factor $1/N$ with $1/(N-1)$.

Derivation. Equation (39) follows directly from Eq. (24). For entity removal,

$$\sum_{j\neq i}\mathbf{u}_j^{(\ell)} = N\boldsymbol{\mu}^{(\ell)} - \mathbf{u}_i^{(\ell)}. \tag{42}$$

Substituting this expression into the mean after removal and subtracting $\boldsymbol{\mu}^{(\ell)}$ gives Eq. (41).

Equations (39) and (41) clarify the difference between replacement and entity removal. Replacement changes only the representation of entity i while keeping the same entities and normalization. Entity removal instead changes both the set of entities and the normalization used to compute the mean. This comparison does not imply that replacement always performs better than removal; it formalizes the difference examined by the *Remove Entity* variant in the ablation study.

B.6 Permutation Equivariance of the Response

The response construction does not depend on the order in which the entities are stored. The mappings used for each entity are shared, and the mean of the transformed representations does not change when the entities are reordered.

Proposition (Permutation equivariance). Let $\mathbf{P}$ be any permutation matrix acting on the entity dimension. Assume that the history representations and the representations of the entities are permuted consistently, while the shared mappings ϕ, ψ, and ρ remain unchanged. Define

$$\mathcal{R}\left(\tilde{\mathbf{H}}^{(\ell)}\right) = [\mathbf{r}_1^{(\ell)}; \ldots; \mathbf{r}_N^{(\ell)}]. \tag{43}$$

Then

$$\mathcal{R}\left(\mathbf{P}\tilde{\mathbf{H}}^{(\ell)}\right) = \mathbf{P}\mathcal{R}\left(\tilde{\mathbf{H}}^{(\ell)}\right). \tag{44}$$

Proof. The shared mapping ϕ is applied in the same way to every entity, so permuting the entities permutes the transformed representations in the same way. Their mean $\frac{1}{N}\sum_j \mathbf{u}_j^{(\ell)}$ is unchanged by the permutation. For each entity, the information from the other entities also contains the same transformed representations after reordering, with only their order changed. Since ψ is shared across entities, the references are permuted in the same way as the entities.

The representation obtained before replacement is unchanged by the permutation because it is computed from the same mean. After replacing one entity, the corresponding representation follows the permutation of that entity. The difference between the two evaluations therefore follows the same permutation, which gives Eq. (44).

This property concerns reordering the same entities and does not imply that responses can be exchanged between different entities. When the complete model is reordered, the entity embeddings used to incorporate the response are permuted together with their corresponding entities.

B.7 Scope of the Interpretation

The analysis above describes the response produced by replacing the representation of one entity while keeping the other entities, normalization, and function unchanged. It does not identify causal effects between entities. The reference $\mathbf{b}_i^{(\ell)}$ is constructed within ERST, and $\mathbf{r}_i^{(\ell)}$ compares two representations that differ only in the representation assigned to entity i. The analysis therefore concerns the mathematical properties of the response and how it changes with the representation of the entity and the other entities.

C Interpretation of Using the Response

This section examines how the response is incorporated into the representation in ERST. Appendix B explains how the response is obtained from replacement. Here, we examine how information about the entity, time, and layer affects the use of the response in the update. The analysis describes the roles of $\mathbf{g}_i^{(\ell)}$ and $\boldsymbol{\alpha}_i^{(\ell)}$ in this process and does not establish a global stability result.

C.1 Using Entity and Time Information

Recall that

$$\mathbf{g}_i^{(\ell)} = \sigma(\mathbf{W}_g \,\mathrm{LN}\,(\mathbf{e}_i + \boldsymbol{\tau}_t + \mathbf{d}_\ell) + \mathbf{b}_g) \in (0,1)^D. \tag{45}$$

Let

$$\mathbf{p}_i^{(\ell)} = g_r\left(\mathbf{r}_i^{(\ell)}\right). \tag{46}$$

The response used in the update can then be written as

$$\bar{\mathbf{r}}_i^{(\ell)} = \mathbf{p}_i^{(\ell)} \odot \mathbf{g}_i^{(\ell)} = \mathrm{Diag}\left(\mathbf{g}_i^{(\ell)}\right)\mathbf{p}_i^{(\ell)}. \tag{47}$$

Equation (47) shows that the projected response is adjusted using information about the entity, time, and layer before it is incorporated into the representation.

For example, suppose two entities have the same projected response $\mathbf{p}$ at the same time and layer. Their responses used in the update satisfy

$$\bar{\mathbf{r}}_i^{(\ell)} - \bar{\mathbf{r}}_j^{(\ell)} = \left[\mathrm{Diag}(\mathbf{g}_i^{(\ell)}) - \mathrm{Diag}(\mathbf{g}_j^{(\ell)})\right] \mathbf{p}. \tag{48}$$

Thus, the same projected response can contribute differently to the representations of different entities because $\mathbf{g}_i^{(\ell)}$ depends on information about the corresponding entity.

C.2 Response in the Update

The response is combined with the updated representation as

$$\mathbf{z}_i^{(\ell)} = \left[\tilde{\mathbf{h}}_i^{(\ell)} \| \bar{\mathbf{r}}_i^{(\ell)}\right], \tag{49}$$

from which ERST computes

$$\mathbf{v}_i^{(\ell)} = g_u(\mathbf{z}_i^{(\ell)}), \tag{50}$$

and

$$\boldsymbol{\alpha}_i^{(\ell)} = \sigma\Big(\mathbf{W}_\alpha \mathbf{z}_i^{(\ell)} + \mathbf{b}_\alpha\Big) \in (0,1)^D. \tag{51}$$

Before layer normalization, the representation is updated as

$$\tilde{\mathbf{h}}_i^{(\ell)} + \boldsymbol{\alpha}_i^{(\ell)} \odot \mathbf{v}_i^{(\ell)}. \tag{52}$$

The sigmoid terms in $\mathbf{g}_i^{(\ell)}$ and $\boldsymbol{\alpha}_i^{(\ell)}$ also give the following bounds. For any p-norm with $1 \le p \le \infty$,

$$\left\|\bar{\mathbf{r}}_i^{(\ell)}\right\|_p \le \left\|\mathbf{p}_i^{(\ell)}\right\|_p, \tag{53}$$

and

$$\left\|\boldsymbol{\alpha}_i^{(\ell)} \odot \mathbf{v}_i^{(\ell)}\right\|_p \le \left\|\mathbf{v}_i^{(\ell)}\right\|_p. \tag{54}$$

Derivation. Every coordinate of $\mathbf{g}_i^{(\ell)}$ and $\boldsymbol{\alpha}_i^{(\ell)}$ lies in $(0,1)$ because both are obtained using sigmoid functions. Therefore,

$$\left|p_{i,k}^{(\ell)} g_{i,k}^{(\ell)}\right| \le \left|p_{i,k}^{(\ell)}\right| \tag{55}$$

and

$$\left|\alpha_{i,k}^{(\ell)} v_{i,k}^{(\ell)}\right| \le \left|v_{i,k}^{(\ell)}\right| \tag{56}$$

for every coordinate k. Applying any p-norm gives Eqs. (53)–(54).

These bounds apply only to the two element-wise multiplications above. They do not establish stability of the complete forecasting model because the learned functions, residual update, and layer normalization also affect the resulting representation. The bounds describe how $\mathbf{g}_i^{(\ell)}$ and $\boldsymbol{\alpha}_i^{(\ell)}$ affect the use of the response in the update.

C.3 Summary of Using the Response

The use of the response in each interaction layer can be summarized as

$$\begin{aligned}
&\mathbf{r}_i^{(\ell)} \xrightarrow{g_r} \mathbf{p}_i^{(\ell)} \xrightarrow{\mathbf{g}_i^{(\ell)}} \bar{\mathbf{r}}_i^{(\ell)}, \\
&\mathbf{z}_i^{(\ell)} = [\tilde{\mathbf{h}}_i^{(\ell)} \| \bar{\mathbf{r}}_i^{(\ell)}], \\
&\mathbf{v}_i^{(\ell)} = g_u(\mathbf{z}_i^{(\ell)}), \\
&\mathbf{h}_i^{(\ell)} = \mathrm{LN}\Big(\tilde{\mathbf{h}}_i^{(\ell)} + \boldsymbol{\alpha}_i^{(\ell)} \odot \mathbf{v}_i^{(\ell)}\Big).
\end{aligned} \tag{57}$$

The vector $\mathbf{g}_i^{(\ell)}$ determines how the response is used according to information about the entity, time, and layer, while $\boldsymbol{\alpha}_i^{(\ell)}$ determines how much the resulting vector contributes to the representation update. The ablation study examines this part of ERST by keeping the response and removing these operations from the update.

D Computational Complexity of ERST

ERST does not construct an $N \times N$ matrix of relations between pairs of entities or reconstruct all entity representations for every replacement. This section describes the computational cost with respect to the number of entities.

D.1 Shared Statistics

Once $\mathbf{u}_i^{(\ell)} = \phi(\tilde{\mathbf{h}}_i^{(\ell)})$ is computed for all entities, the mean is obtained as

$$\boldsymbol{\mu}^{(\ell)} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{u}_j^{(\ell)}. \tag{58}$$

The information from the other entities used to construct the reference can then be obtained without computing a separate sum for every entity:

$$\boldsymbol{\mu}_{-i}^{(\ell)} = \frac{N\boldsymbol{\mu}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N-1}. \tag{59}$$

After $\mathbf{b}_i^{(\ell)}$ and $\mathbf{u}_{i,b}^{(\ell)}$ are computed, the mean after replacing the representation of entity i is

$$\boldsymbol{\mu}_{\mathrm{rep},i}^{(\ell)} = \boldsymbol{\mu}^{(\ell)} + \frac{\mathbf{u}_{i,b}^{(\ell)} - \mathbf{u}_i^{(\ell)}}{N}. \tag{60}$$

Thus, the quantities required for each entity are obtained from the same mean by changing only the contribution associated with that entity, without reconstructing the representations of all entities for every replacement.

D.2 Complexity with Respect to the Number of Entities

Assume that the widths of the mappings ϕ, ψ, ρ, g_r, and g_u are $\mathcal{O}(D)$, where D is the representation dimension. One ERST interaction layer requires

$$\mathcal{O}(ND^2) \tag{61}$$

computation and $\mathcal{O}(ND)$ storage for the representations required during the interaction, excluding parameter storage. For fixed D, the computational cost is therefore linear in the number of entities N.

Derivation. Computing the mean and the quantities in Eq. (59) and Eq. (24) for all entities requires $\mathcal{O}(ND)$ operations. The mappings used by ERST are applied to the representations of N entities. With width $\mathcal{O}(D)$, each mapping requires $\mathcal{O}(D^2)$ computation per entity, resulting in $\mathcal{O}(ND^2)$ computation overall. The entity representations, references, responses, and vectors used in the update require $\mathcal{O}(ND)$ storage. ERST does not require an $N \times N$ matrix of relations between entities.

A direct implementation that reconstructs all entity representations for each of the N replacements would require $\mathcal{O}(N^2D)$ representation entries to be stored or processed. Equations (59) and (24) avoid this reconstruction by obtaining the required quantities from the same mean. This analysis describes the complexity of the ERST interaction and complements the runtime and parameter results reported in the main paper. It does not imply that all methods using graphs have quadratic complexity; it only describes how the computation in ERST scales with the number of entities.

E Detailed Experimental Setup

E.1 Datasets and Forecasting Protocols

We evaluate ERST on 12 spatiotemporal forecasting datasets from traffic sensing, urban mobility, air quality, wind power, and mobile communication systems. Table 6 summarizes the statistics of the processed datasets used in our experiments. Here, *Time Steps* denotes the number of temporal observations after preprocessing, and *#Entities* denotes the number of entities used for forecasting.

Table 6: Statistics and forecasting protocols of the evaluated datasets. The reported statistics correspond to the processed data used in our experiments. For the urban mobility datasets, we report the inflow forecasting setting.

Domain	Dataset	Time Steps	#Entities	Interval	Target Variable	Input→Output
Traffic Sensing	PEMS03	26,208	358	5 min	Traffic flow	12→12
Traffic Sensing	PEMS04	16,992	307	5 min	Traffic flow	12→12
Traffic Sensing	PEMS07	28,224	883	5 min	Traffic flow	12→12
Traffic Sensing	PEMS08	17,856	170	5 min	Traffic flow	12→12
Wind Power	SDWPF	35,280	134	10 min	Active power (Patv)	12→12
Air Quality	KnowAir	11,688	184	3 h	$PM_{2.5}$	12→12
Mobile Communication	Milan	1,464	400	1 h	Internet traffic	12→12
Traffic Sensing	LargeST-GLA	35,040	3,834	15 min	Traffic flow	12→12
Traffic Sensing	LargeST-CA	35,040	8,600	15 min	Traffic flow	12→12
Urban Mobility	NYCTaxi	17,520	75	30 min	Inflow	6→1
Urban Mobility	CHIBike	4,416	270	30 min	Inflow	6→1
Urban Mobility	T-Drive	3,600	1,024	60 min	Inflow	6→1

Traffic sensing datasets. PEMS03, PEMS04, PEMS07, and PEMS08 are traffic forecasting benchmarks derived from the California Performance Measurement System (PeMS) (Guo et al., 2022). They contain 358, 307, 883, and 170 traffic sensors, respectively, with measurements at 5-minute intervals. The benchmark periods correspond to September–November 2018 for PEMS03, January–February 2018 for PEMS04, May–August 2017 for PEMS07, and July–August 2016 for PEMS08.

Wind power and air quality datasets. SDWPF is a wind power forecasting dataset containing measurements from 134 wind turbines at 10-minute intervals (Zhou et al., 2024). We use active power (*Patv*) as the forecasting target. The processed sequence used in our experiments contains 35,280 time steps, corresponding to the 245-day forecasting benchmark derived from the SDWPF data.

KnowAir is an air quality dataset constructed from observations over 184 Chinese cities from January 2015 to December 2018 (Wang et al., 2020). The observations are provided at 3-hour intervals, resulting in 11,688 time steps, and we use $PM_{2.5}$ concentration as the forecasting target.

Mobile communication dataset. The Milan telecommunications dataset was released through the Telecom Italia Big Data Challenge and contains mobile communication activity collected over Milan during November and December 2013 (Barlacchi et al., 2015). The original dataset is provided on a spatial grid at 10-minute intervals. For our forecasting setting, we use the Internet activity variable after spatial and temporal aggregation, resulting in a 20×20 grid (400 entities), 1-hour intervals, and 1,464 time steps (Zhang et al., 2019).

LargeST datasets. LargeST is a traffic forecasting benchmark constructed from Caltrans PeMS measurements and contains several subsets with different numbers of entities (Liu et al., 2023). We evaluate ERST on the Greater Los Angeles (GLA) subset with 3,834 sensors and the California (CA) system with 8,600 sensors. Following the 2019 LargeST benchmark, traffic flow observations are aggregated at 15-minute intervals, producing 35,040 time steps for each dataset.

Urban mobility datasets. We further evaluate ERST on NYCTaxi, CHIBike, and T-Drive using the forecasting protocol adopted by PDFormer (Jiang et al., 2023a). NYCTaxi contains 75 cells with 17,520 observations at 30-minute intervals, CHIBike contains 270 cells with 4,416 observations at 30-minute intervals, and T-Drive contains 1,024 cells with 3,600 observations at 60-minute intervals. The corresponding observation periods are 2014 for NYCTaxi, July–September 2020 for CHIBike, and February–June 2015 for T-Drive.

Data splitting and normalization. For PEMS03, PEMS04, PEMS07, PEMS08, SDWPF, KnowAir, Milan, LargeST-CA, and LargeST-GLA, we chronologically split the observations into training, validation, and test sets with a ratio of 6:2:2. For NYCTaxi, CHIBike, and T-Drive, we

follow the forecasting protocol used by PDFormer (Jiang et al., 2023a) and use a 7:1:2 split. For all datasets, the statistics used for normalization are computed from the training set only and then applied to the validation and test sets.

E.2 Baseline Methods

We compare ERST with 23 forecasting methods, including statistical and recurrent methods, methods using graphs, Transformers, or MLPs, and recent methods designed for spatiotemporal forecasting with many entities. For PEMS03, PEMS04, PEMS07, PEMS08, KnowAir, SDWPF, and Milan, all 23 methods are evaluated. For NYCTaxi, CHIBike, and T-Drive, we evaluate 14 methods whose released implementations can be used with the forecasting protocol adopted for these datasets without substantial changes to the model or data format.

- **HA** (Pan et al., 2012) predicts future values using historical averages from corresponding temporal periods.
- **SVR** (Drucker et al., 1996) uses kernel functions to model nonlinear relationships between historical observations and future values.
- **LSTM** (Hochreiter & Schmidhuber, 1997) models temporal dependencies using recurrent units with gating mechanisms for sequential observations.
- **DCRNN** (Li et al., 2018) combines diffusion graph convolution with a recurrent encoder–decoder to model spatial and temporal dependencies.
- **STGCN** (Yu et al., 2018) combines graph convolution for spatial modeling with temporal convolution for temporal modeling.
- **Graph WaveNet** (Wu et al., 2019) learns an adaptive adjacency structure and combines graph convolution with dilated causal temporal convolution.
- **AGCRN** (Bai et al., 2020) uses adaptive graph convolution and recurrent modeling to learn spatial dependencies from data.
- **MTGNN** (Wu et al., 2020) learns a graph structure from multivariate observations and combines graph propagation with temporal convolution.
- **STGODE** (Fang et al., 2021) uses neural ordinary differential equations to model spatiotemporal dynamics over graph-structured observations.
- **FEDformer** (Zhou et al., 2022) combines time-series decomposition with frequency-domain Transformer components for temporal forecasting.
- **STID** (Shao et al., 2022) uses spatial and temporal identity embeddings with an MLP architecture without explicitly constructing relations between pairs of entities.
- **MegaCRN** (Jiang et al., 2023b) combines memory-based representations with graph convolutional recurrent modeling for spatiotemporal forecasting.
- **PatchTST** (Nie et al., 2023) represents temporal sequences as patches and applies a Transformer encoder to model temporal dependencies.
- **iTransformer** (Liu et al., 2024) treats individual variables as tokens and applies attention across variables to model their dependencies.
- **STPGNN** (Kong et al., 2024) identifies pivotal nodes and separately models relations involving pivotal and non-pivotal nodes.
- **DFDGCN** (Li et al., 2024) learns dynamic graph structures in the frequency domain while modeling spatial dependencies.
- **SOFTS** (Han et al., 2024b) aggregates multiple series into one representation and uses the resulting information to update individual series through an MLP architecture.
- **TWIST** (Wang et al., 2025) uses temporal-window and sparse attention to model spatial and temporal dependencies.
- **PatchSTG** (Fang et al., 2025) partitions large sensor networks into spatial patches and applies attention within and across patches.
- **LightST** (Zhang et al., 2025) transfers spatial and temporal information from a teacher model to a lightweight forecasting model through knowledge distillation.

- **STDN** (Cao et al., 2025) combines dynamic spatial modeling with trend–seasonality decomposition for traffic forecasting.
- **RAST** (Ruan et al., 2026) uses retrieved historical patterns as additional information for spatiotemporal forecasting.
- **FaST** (Zhao et al., 2026) uses a mixture-of-experts architecture for forecasting on large spatiotemporal graphs.

Table 7: Default implementation settings of ERST.

Setting	Value
Latent dimension D	64
Interaction layers L	2
Dropout rate	0.1
Batch size	32
Optimizer	AdamW
Initial learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Maximum epochs	100
Early-stopping patience	15 epochs
Model selection	Lowest validation MAE
Normalization statistics	Training split only
Spatial structure	No predefined graph / adjacency matrix

E.3 Implementation Details

We use the same implementation settings across datasets unless otherwise specified. The architecture and optimization hyperparameters are fixed, while the number of entities, temporal resolution, input length, and forecasting horizon follow the settings of each dataset. Table 7 summarizes the default configuration used for ERST.

E.4 Evaluation Metrics

Following standard evaluation protocols in spatiotemporal forecasting (Jiang et al., 2023a; Li et al., 2018; Yu et al., 2018; Shao et al., 2022), we evaluate forecasting performance using mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Lower values indicate better forecasting performance for all three metrics.

F Full Forecasting Results

This section reports the complete forecasting results for the evaluated datasets and methods. The main text presents selected comparisons, while the tables below report the complete results using the same forecasting protocols. We report MAE, RMSE, and MAPE for the PeMS datasets, and MAE and RMSE for KnowAir, SDWPF, and Milan. Lower values indicate better forecasting performance.

F.1 Traffic Sensing

Table 8 reports the complete results on PEMS03, PEMS04, PEMS07, and PEMS08. ERST obtains the lowest MAE, RMSE, and MAPE on all four datasets.

F.2 Environmental, Energy, and Mobile Communication Forecasting

Table 9 reports the complete results on KnowAir, SDWPF, and Milan. These datasets differ in the number of entities, temporal resolution, and forecasting target, providing evaluations across air quality, wind power, and mobile communication settings. ERST obtains the lowest MAE and RMSE on all three datasets.

Table 8: Full forecasting results on PEMS03, PEMS04, PEMS07, and PEMS08. Lower values are better. The best results are shown in bold and the second-best results are underlined.

	PEMS03			PEMS04			PEMS07			PEMS08		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	32.33	53.54	32.50	38.68	58.55	26.46	47.53	74.61	22.75	36.46	54.19	23.64
SVR	19.47	32.37	18.76	25.05	38.34	17.65	28.42	44.13	12.82	20.27	31.08	12.57
LSTM	18.67	32.97	18.08	22.48	37.86	14.86	30.31	55.06	13.19	21.02	34.01	12.44
DCRNN	17.54	29.70	16.89	24.01	37.59	15.75	24.54	38.02	10.70	18.47	28.57	11.65
STGCN	17.05	29.42	15.64	22.17	34.90	14.03	24.59	38.78	11.08	18.05	27.89	11.51
GWNet	14.61	25.23	15.44	19.35	31.71	13.30	21.22	34.11	9.07	15.06	24.85	9.51
AGCRN	16.63	28.36	15.62	20.03	33.32	13.12	21.13	34.69	8.85	20.03	33.23	13.12
MTGNN	14.91	25.43	15.92	19.17	30.47	13.69	21.02	33.89	9.11	15.06	23.82	10.10
STGODE	16.73	27.12	16.69	21.70	33.39	13.89	23.84	36.55	10.10	18.09	27.23	11.75
FEDformer	19.38	30.14	20.68	26.41	40.94	20.36	26.23	40.10	11.75	20.51	32.31	12.74
STID	15.28	26.85	16.36	19.82	32.04	13.58	21.61	32.79	9.23	15.65	25.86	10.33
MegaCRN	14.70	26.20	15.27	18.81	30.62	13.30	19.72	32.80	8.36	14.58	23.48	10.11
PatchTST	21.04	33.45	18.95	27.40	42.72	18.15	30.36	47.13	12.96	21.91	34.30	13.14
iTransformer	18.02	28.79	16.35	24.38	38.78	16.07	26.40	41.67	11.11	22.84	35.46	13.83
STPGNN	17.95	28.53	17.26	24.59	37.95	16.54	26.04	39.91	11.45	18.86	29.38	12.23
DFDGCN	14.62	24.48	14.85	18.79	30.14	13.32	20.35	33.40	8.79	14.32	23.28	9.79
SOFTS	18.01	28.64	16.37	24.61	39.00	16.23	26.15	41.28	11.01	19.81	31.41	11.82
TWIST	15.11	24.76	15.87	18.51	29.92	12.90	19.70	32.70	8.44	14.21	23.16	9.36
PatchSTG	14.99	24.75	15.14	18.32	30.30	12.36	20.37	33.82	8.11	14.29	24.08	9.66
LightST	15.14	24.72	14.29	19.12	31.04	12.73	21.18	34.30	9.08	15.45	24.61	9.81
STDN	15.36	26.95	17.03	18.50	30.74	12.28	20.37	33.65	9.40	14.56	23.65	9.87
RAST	15.89	25.34	17.14	20.10	31.79	13.57	21.28	34.21	9.27	15.98	25.16	10.50
FaST	15.41	25.10	15.26	19.89	32.77	13.27	21.22	36.11	8.86	14.92	25.47	9.52
ERST (Ours)	**14.28**	**22.20**	**13.59**	**18.18**	**28.06**	**11.74**	**19.51**	**31.96**	**8.02**	**14.08**	**22.19**	**8.65**

Table 9: Full forecasting results on KnowAir, SDWPF, and Milan Internet. Lower values are better. Best results are shown in bold and second-best results are underlined.

	KnowAir		SDWPF		Milan Internet	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	28.56	35.72	352.60	435.03	1111.96	1663.80
SVR	15.57	24.49	112.76	179.32	554.60	915.29
LSTM	16.85	25.11	105.24	179.70	886.15	1323.56
DCRNN	16.97	24.30	216.93	337.54	250.67	473.19
STGCN	16.66	25.40	119.11	189.50	617.31	745.93
GWNet	14.89	22.18	96.88	165.85	404.57	693.90
AGCRN	15.53	23.83	100.39	169.54	690.69	1117.11
MTGNN	14.85	23.36	94.19	165.75	291.69	550.47
STGODE	14.88	22.77	101.19	164.23	432.99	729.87
FEDformer	17.48	26.62	107.04	168.07	290.53	575.69
STID	15.04	23.71	102.78	178.09	423.80	684.63
MegaCRN	14.45	22.25	98.02	176.74	346.16	593.93
PatchTST	15.79	24.95	96.72	171.20	290.53	575.69
iTransformer	15.31	24.31	95.12	166.37	239.26	506.14
STPGNN	14.33	23.03	94.98	167.64	295.56	529.91
DFDGCN	14.46	23.52	120.76	215.55	234.77	415.90
SOFTS	15.34	24.37	97.14	167.15	240.51	511.58
TWIST	19.92	29.16	140.47	208.34	448.83	942.40
PatchSTG	14.19	22.45	98.66	165.30	407.21	773.71
LightST	14.44	22.58	95.89	163.88	292.85	534.05
STDN	14.30	22.03	105.09	174.27	440.45	874.95
RAST	15.19	23.83	98.56	174.30	329.85	637.85
FaST	14.84	23.74	95.32	168.14	203.62	420.83
ERST (Ours)	**13.98**	**20.50**	**93.08**	**131.20**	**201.52**	**409.95**

F.3 URBAN MOBILITY FORECASTING

We further report the complete results on NYCTaxi, T-Drive, and CHIBike. Following the forecasting protocol described in Section 4.1, all methods use six historical observations to predict the next step with the same training, validation, and test splits. We evaluate both inflow and outflow forecasting using MAE, RMSE, and MAPE.

Among the methods considered in the overall evaluation, we include 14 methods whose released implementations can be used with this forecasting protocol without substantial changes to the model or data format. These methods are STGCN, Graph WaveNet, AGCRN, MTGNN, STGODE, FEDformer, STID, PatchTST, iTransformer, STPGNN, SOFTS, TWIST, RAST, and FaST. Table 10 reports the complete results.

Table 10: Full forecasting results on NYCTaxi, T-Drive, and CHIBike. In and Out denote inflow and outflow forecasting, respectively. Lower values are better. The best results are shown in bold and the second-best results are underlined.

	NYCTaxi						T-Drive						CHIBike					
	In			Out			In			Out			In			Out		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	14.37	23.87	14.35	12.82	20.06	14.34	22.12	38.65	17.75	21.75	38.16	17.38	4.23	5.99	31.26	4.21	5.82	30.82
GWNet	14.31	23.79	14.19	12.28	19.61	13.68	19.55	36.15	17.18	19.55	36.19	15.93	4.15	5.91	31.15	4.10	5.69	30.69
AGCRN	14.93	25.50	14.02	13.14	25.43	14.01	24.77	68.22	15.28	25.15	69.18	15.21	4.06	5.87	30.60	4.10	6.03	31.06
MTGNN	14.17	23.61	14.10	12.37	20.10	13.36	20.41	36.93	15.57	19.77	35.63	15.32	3.60	5.49	**24.34**	3.61	5.54	24.81
STGODE	14.62	25.44	14.79	12.83	20.20	14.39	21.51	38.21	17.57	22.70	40.28	18.50	4.16	5.92	31.16	4.12	5.69	30.72
FEDformer	20.46	33.59	20.76	17.49	28.24	20.19	31.12	55.71	23.98	31.52	56.39	24.23	5.36	8.00	39.23	5.10	7.73	37.25
STID	14.14	23.59	14.07	11.97	18.87	13.85	17.90	31.21	14.67	18.00	31.34	14.77	4.04	5.78	30.24	4.20	6.12	31.52
PatchTST	16.32	30.96	28.93	15.33	29.06	27.15	29.26	64.50	34.29	29.36	64.67	34.63	8.40	11.70	56.82	8.47	11.45	56.65
iTransformer	19.88	32.93	19.18	17.27	27.75	18.84	33.13	63.17	23.13	49.91	89.99	36.68	7.27	10.92	49.81	6.81	10.53	45.71
STPGNN	20.52	34.05	18.79	18.10	29.22	20.32	36.21	73.14	24.38	34.24	67.42	22.53	4.32	6.69	28.82	4.41	6.85	30.67
SOFTS	20.39	33.49	19.65	13.80	25.08	24.51	24.89	55.19	30.13	25.07	55.55	30.16	5.41	7.75	39.07	4.43	8.07	53.19
TWIST	13.65	23.00	13.48	11.64	18.84	13.43	17.96	31.90	14.27	17.79	31.79	14.73	3.86	5.91	25.65	3.61	5.47	25.80
RAST	14.64	24.63	13.99	12.53	20.12	14.02	22.95	44.33	17.00	22.91	43.66	16.97	3.82	5.94	24.76	3.85	6.03	25.80
FaST	15.07	25.10	15.36	13.02	20.60	15.14	20.45	37.56	16.78	20.35	36.95	16.70	4.56	6.50	34.57	4.28	6.09	33.41
ERST (Ours)	**13.43**	**22.69**	**12.97**	**11.54**	**18.57**	**12.81**	**16.56**	**29.05**	**13.44**	**16.56**	**29.10**	**13.54**	**3.52**	**5.36**	24.35	**3.42**	**5.25**	**23.81**

ERST obtains the lowest error in 17 of the 18 reported dataset–direction–metric combinations. On NYCTaxi, ERST obtains the lowest MAE, RMSE, and MAPE for both inflow and outflow forecasting. The same result is observed on T-Drive, where ERST obtains the lowest value for all six reported metrics. On CHIBike, ERST obtains the lowest values for all metrics except inflow MAPE, where ERST achieves 24.35 and MTGNN achieves 24.34. These results show that the same ERST interaction can be used for both inflow and outflow forecasting without changing the interaction for each dataset or forecasting direction.

Table 11: Full forecasting results on LargeST-CA and LargeST-GLA. Lower values are better. The best results are shown in bold and the second-best results are underlined.

	LargeST-CA			LargeST-GLA		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STID	18.61	32.00	14.25	21.03	34.68	14.46
PatchTST	46.32	75.65	32.37	51.07	82.65	30.54
SOFTS	33.12	56.29	24.66	34.72	59.16	22.17
TWIST	24.88	36.73	18.43	34.88	50.20	28.95
PatchSTG	**17.65**	30.25	12.85	**19.32**	32.63	11.48
RAST	29.30	49.13	21.71	31.65	52.20	20.76
FaST	23.95	44.19	17.25	24.87	43.40	15.33
ERST (Ours)	18.03	**29.67**	**12.13**	19.80	**32.13**	**11.23**

F.4 LARGEST FORECASTING

Table 11 reports the complete results on LargeST-CA and LargeST-GLA across MAE, RMSE, and MAPE. Because these datasets contain 8,600 and 3,834 entities, respectively, we compare ERST with seven forecasting methods: STID, PatchTST, SOFTS, TWIST, PatchSTG, RAST, and FaST.

PatchSTG achieves the lowest MAE on both LargeST-CA and LargeST-GLA, while ERST achieves the second lowest MAE. ERST achieves the lowest RMSE and MAPE on both datasets. Across the six reported measures, ERST obtains the lowest value on four measures and the second lowest value on the remaining two.

G LONG-HORIZON FORECASTING

We further evaluate ERST with longer forecasting horizons on PEMS04 and PEMS08. We use horizons of 36, 48, and 60 steps, corresponding to 3, 4, and 5 hours, respectively, at 5-minute intervals. These experiments examine forecasting performance as the horizon increases beyond the standard 12-step setting. The Average columns report the arithmetic mean over the three horizons. Lower values indicate better performance.

Table 12: Long-horizon forecasting results on PEMS04. H denotes the forecasting horizon in steps. Lower values are better. The best results are shown in bold and the second-best results are underlined.

	$H = 36$			$H = 48$			$H = 60$			Average		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	28.98	45.92	18.52	31.94	50.54	20.34	32.62	51.83	20.28	31.18	49.43	19.71
AGCRN	21.47	35.25	14.27	22.13	36.59	14.62	22.76	37.11	15.38	22.12	36.32	14.76
MTGNN	21.60	33.73	15.12	22.24	34.69	16.29	22.97	35.69	16.69	22.27	34.70	16.03
STGODE	25.36	38.75	17.99	27.61	41.51	21.77	27.64	41.61	21.70	26.87	40.62	20.49
FEDformer	33.35	49.44	26.52	36.82	53.49	33.09	38.87	55.72	36.24	36.35	52.88	31.95
STID	20.63	33.12	14.18	21.58	34.59	15.75	21.58	35.18	15.51	21.26	34.30	15.15
PatchTST	45.73	71.50	31.91	52.67	81.00	40.74	56.79	86.44	50.36	51.73	79.65	41.00
iTransformer	28.36	43.59	22.34	30.44	46.97	23.54	31.07	47.90	24.00	29.96	46.15	23.29
STPGNN	34.59	52.57	24.27	38.43	56.97	30.30	40.08	60.52	29.13	37.70	56.69	27.90
SOFTS	36.68	58.16	26.10	40.58	63.27	31.13	43.77	67.05	36.19	40.34	62.83	31.14
TWIST	20.42	32.86	14.13	21.21	34.30	14.80	22.19	35.64	15.82	21.27	34.27	14.92
PatchSTG	20.88	33.85	14.75	22.28	37.17	14.20	23.26	38.31	14.58	22.14	36.44	14.51
RAST	22.63	35.37	16.86	23.70	37.20	17.30	24.15	37.81	17.98	23.49	36.79	17.38
FaST	23.63	38.76	16.53	24.63	39.96	18.32	25.52	41.02	20.24	24.59	39.91	18.36
ERST (Ours)	**20.02**	**30.68**	**13.19**	**20.75**	**31.85**	**13.75**	**21.31**	**32.67**	**14.51**	**20.69**	**31.73**	**13.82**

G.1 PEMS04

Table 12 reports the results on PEMS04. ERST obtains the lowest MAE, RMSE, and MAPE at all three forecasting horizons and the lowest average values across the three horizons.

G.2 PEMS08

Table 13 reports the results on PEMS08. ERST obtains the lowest RMSE and MAPE at all three forecasting horizons. It also obtains the lowest MAE at $H = 36$ and $H = 60$, while STID obtains a slightly lower MAE at $H = 48$. Averaged across the three horizons, ERST obtains the lowest values for all three metrics.

Across PEMS04 and PEMS08, ERST is evaluated at forecasting horizons from 36 to 60 steps. On PEMS04, ERST obtains the lowest value for all horizon-specific and averaged metrics. On PEMS08, ERST obtains the lowest value for 8 of the 9 horizon-specific metrics and for all three averaged metrics. The only exception is MAE at $H = 48$, where ERST obtains the second-lowest value. These results show that the same ERST interaction can be used when the forecasting horizon is increased from 36 to 60 steps.

Table 13: Long-horizon forecasting results on PEMS08. H denotes the forecasting horizon in steps. Lower values are better. The best results are shown in bold and the second-best results are underlined.

	$H = 36$			$H = 48$			$H = 60$			Average		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	22.93	35.82	14.46	24.88	41.09	16.95	26.54	42.38	17.68	24.78	39.76	16.36
AGCRN	18.75	30.31	12.11	25.65	39.08	17.36	20.99	33.74	14.67	21.80	34.38	14.71
MTGNN	17.14	27.30	11.96	18.01	28.58	12.86	18.46	29.01	13.50	17.87	28.30	12.77
STGODE	22.44	34.51	15.27	23.43	36.08	16.43	24.71	37.76	17.29	23.53	36.12	16.33
FEDformer	31.11	46.76	20.79	33.94	50.09	23.43	37.51	54.03	25.73	34.19	50.29	23.32
STID	16.29	26.85	10.67	**16.72**	27.57	11.37	17.41	28.32	11.84	16.81	27.58	11.29
PatchTST	36.33	57.46	22.18	40.37	63.28	26.29	43.14	67.30	30.49	39.95	62.68	26.32
iTransformer	25.91	41.86	18.54	30.47	48.98	21.78	30.45	49.60	22.47	28.94	46.81	20.93
STPGNN	25.32	39.37	17.24	27.32	42.33	18.62	27.59	42.51	19.55	26.74	41.40	18.47
SOFTS	28.76	45.52	18.07	31.26	48.93	21.07	33.74	51.97	24.31	31.25	48.81	21.15
TWIST	16.70	26.85	11.41	17.35	27.95	12.35	17.85	28.90	12.09	17.30	27.90	11.95
PatchSTG	16.69	28.75	10.86	17.58	29.35	11.88	18.20	31.05	12.88	17.49	29.72	11.87
RAST	18.71	29.45	13.08	19.62	30.79	14.75	20.05	31.47	14.40	19.46	30.57	14.08
FaST	18.43	31.39	12.26	20.09	34.20	13.80	21.40	35.88	15.32	19.97	33.82	13.79
ERST (Ours)	**16.10**	**25.39**	**10.09**	16.83	**26.45**	**10.77**	**17.38**	**27.30**	**11.10**	**16.77**	**26.38**	**10.65**

H Additional Ablation and Sensitivity Analyses

H.1 Extended Ablation Results

We extend the ablation study in the main paper to ten datasets from traffic sensing, air quality, wind power, urban mobility, and mobile communication. The variants follow the same definitions used in the main text.

Temporal-only keeps temporal modeling for each entity while removing the ERST interaction. *Original Representation* and *Replaced Representation* use either representation alone, whereas *w/o Response* removes the difference between them. *Remove Entity* removes entity i instead of keeping the same entities and replacing only its representation. *w/o Modulation and Gating* keeps the response but removes the operations used to adjust and incorporate it into the representation.

For readability, we group the complete results by domain in Tables 14–16.

Response from Replacement. Across the extended evaluation, *Original Representation* and *Replaced Representation* obtain higher errors than ERST. Similarly, *w/o Response* reduces forecasting performance across the evaluated datasets. These results are consistent with the main-text observation that the difference obtained before and after replacement provides information that is not given by either representation alone.

Replacement versus Entity Removal. *Remove Entity* obtains higher errors than ERST across the reported metrics, except for the tied RMSE on CHIBike-In. Removing an entity changes both the set of entities and the normalization, whereas ERST keeps the same entities and normalization and replaces only the representation of entity i. These results provide an additional comparison between the two operations.

Using the Response. *w/o Modulation and Gating* keeps the response but removes the operations used to adjust and incorporate it into the representation. Its errors are higher than those of ERST across all ten datasets. This result shows that, in addition to constructing the response, how the response is used in the representation update also affects forecasting performance.

Comparison with Temporal Modeling Alone. ERST obtains lower MAE than *Temporal-only* on all ten datasets. Relative to *Temporal-only*, the MAE reduction ranges from 4.05% on Milan to 30.83% on T-Drive. Figure 6 further reports this comparison on six datasets for both MAE

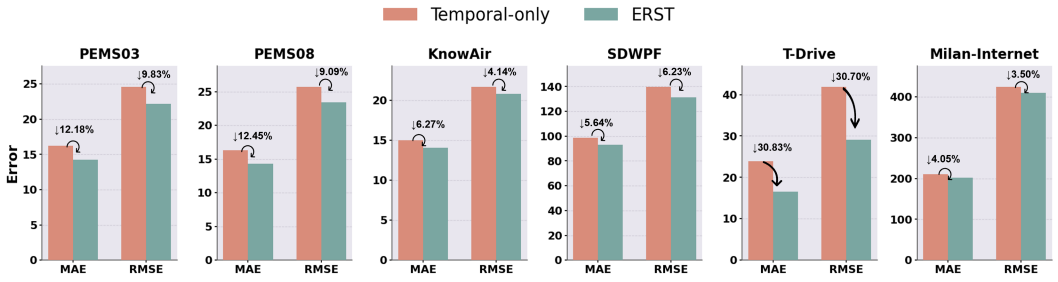


Figure 6: Comparison between *Temporal-only* and ERST on six datasets. Lower MAE and RMSE are better. *Temporal-only* keeps temporal modeling for each entity while removing the ERST interaction.

and RMSE. These results show that the ERST interaction provides forecasting information beyond temporal modeling for each entity.

Table 14: Extended ablation results on PEMS03, PEMS04, PEMS07, and PEMS08. Lower values are better. The best results are shown in bold.

	PEMS03			PEMS04			PEMS07			PEMS08		
Variant	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Temporal-only	16.26	24.62	15.41	22.03	33.33	14.36	23.45	37.18	9.74	16.31	25.74	10.10
Replaced Representation	15.65	23.68	14.73	21.29	32.30	13.87	22.34	36.06	9.18	15.60	24.82	9.81
Original Representation	15.68	23.69	14.70	21.33	32.36	13.84	22.43	36.05	9.36	15.65	24.80	9.81
w/o $\mathbf{s}_i$	14.58	22.88	14.42	18.46	28.14	11.95	20.36	32.28	8.38	14.43	22.42	9.08
w/o $\boldsymbol{\mu}_{-i}$	14.52	22.60	13.68	18.44	28.11	12.09	20.16	32.37	8.38	14.41	22.62	8.80
w/o Response	16.23	24.63	15.32	21.96	33.26	14.34	23.31	37.05	9.81	16.18	25.64	10.17
w/o Modulation and Gating	15.67	23.77	14.95	21.28	32.30	13.88	22.45	36.07	9.44	15.54	24.78	9.60
Remove Entity	14.33	22.69	13.77	18.25	28.07	12.09	19.71	32.00	8.19	14.20	22.40	8.78
ERST	**14.28**	**22.20**	**13.59**	**18.18**	**28.06**	**11.74**	**19.51**	**31.96**	**8.02**	**14.08**	**22.19**	**8.65**

Table 15: Extended ablation results on KnowAir, SDWPF, and Milan. Lower values are better. The best results are shown in bold.

	KnowAir		SDWPF		Milan	
Variant	MAE	RMSE	MAE	RMSE	MAE	RMSE
Temporal-only	15.00	21.73	98.64	139.92	210.02	424.80
Replaced Representation	14.49	21.05	94.65	133.56	215.12	420.88
Original Representation	14.36	21.03	95.31	133.57	223.12	415.52
w/o $\mathbf{s}_i$	14.02	20.72	95.12	133.74	230.15	417.43
w/o $\boldsymbol{\mu}_{-i}$	14.05	20.80	96.33	135.52	268.28	525.55
w/o Response	15.00	21.76	98.98	139.75	206.80	424.04
w/o Modulation and Gating	14.36	21.06	95.39	135.21	215.17	422.83
Remove Entity	14.10	21.33	94.90	135.79	215.61	415.22
ERST	**13.98**	**20.50**	**93.08**	**131.20**	**201.52**	**409.95**

Analysis of Reference Construction. We further examine how the reference $\mathbf{b}_i^{(\ell)}$ is constructed using three variants. First, $\psi = 0$ sets the output of the reference mapping to zero while keeping the remaining model unchanged. This variant examines whether a learned reference contributes to forecasting performance. Second, *w/o* $\mathbf{s}_i$ removes the history representation of entity i from the input to ψ, while *w/o* $\boldsymbol{\mu}_{-i}$ removes the information from the other entities. These two variants examine the contributions of the history of entity i and the representations of the other entities to reference construction.

Table 16: Extended ablation results on T-Drive, NYCTaxi, and CHIBike for inflow forecasting. Lower values are better. The best results are shown in bold.

	T-Drive-In			NYCTaxi-In			CHIBike-In		
Variant	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Temporal-only	23.94	41.92	18.24	16.44	27.46	15.79	3.91	6.01	26.41
Replaced Representation	22.74	39.31	17.72	15.58	25.88	15.10	3.65	5.57	25.06
Original Representation	22.95	39.33	17.79	15.63	26.01	14.99	3.63	5.58	24.49
w/o $\mathbf{s}_i$	17.81	31.30	14.71	13.65	22.98	13.16	3.55	5.40	24.44
w/o $\boldsymbol{\mu}_{-i}$	18.04	31.93	14.94	13.93	23.00	13.30	3.63	5.54	25.45
w/o Response	24.01	41.91	18.44	16.41	27.38	15.71	3.90	5.99	26.27
w/o Modulation and Gating	22.66	39.60	17.95	15.60	25.95	15.00	3.63	5.57	24.78
Remove Entity	18.15	31.22	14.14	13.49	23.50	13.00	3.54	**5.36**	24.60
ERST	**16.56**	**29.05**	**13.44**	**13.43**	**22.69**	**12.97**	**3.52**	**5.36**	**24.35**

To examine whether these results are consistent across different random seeds, we repeat the three variants and ERST using five random seeds. Table 17 reports the mean and standard deviation of MAE and RMSE together with two-sided paired t-tests against ERST using the same random seeds.

Table 17: Five-seed analysis of reference construction. Results are reported as mean±standard deviation over five random seeds. p-values are obtained from two-sided paired t-tests against ERST using the same random seeds. Lower values are better. $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

Dataset	Metric	$\psi = 0$	p-value	w/o $\mathbf{s}_i$	p-value	w/o $\boldsymbol{\mu}_{-i}$	p-value	**ERST**
PEMS03	MAE	14.334 ± 0.034	0.0452^{*}	14.516 ± 0.136	0.0211^{*}	14.690 ± 0.150	0.00380^{**}	$\mathbf{14.242 \pm 0.053}$
	RMSE	22.612 ± 0.088	0.0010^{**}	22.820 ± 0.069	$1.74 \times 10^{-4***}$	22.828 ± 0.137	$8.76 \times 10^{-4***}$	$\mathbf{22.114 \pm 0.087}$
PEMS04	MAE	18.298 ± 0.057	0.0355^{*}	18.466 ± 0.030	$2.60 \times 10^{-6***}$	18.424 ± 0.015	$1.42 \times 10^{-4***}$	$\mathbf{18.202 \pm 0.029}$
	RMSE	28.602 ± 0.172	0.0014^{**}	28.160 ± 0.075	0.0153^{*}	28.108 ± 0.004	0.00416^{**}	$\mathbf{28.026 \pm 0.034}$
PEMS07	MAE	19.628 ± 0.018	0.00999^{**}	20.150 ± 0.126	$2.59 \times 10^{-4***}$	20.072 ± 0.070	$3.11 \times 10^{-4***}$	$\mathbf{19.450 \pm 0.078}$
	RMSE	32.008 ± 0.018	0.0022^{**}	32.196 ± 0.088	0.00173^{**}	32.246 ± 0.088	$8.51 \times 10^{-4***}$	$\mathbf{31.928 \pm 0.019}$
PEMS08	MAE	14.228 ± 0.019	0.0001^{***}	14.410 ± 0.016	$1.61 \times 10^{-5***}$	14.418 ± 0.035	$4.79 \times 10^{-6***}$	$\mathbf{14.074 \pm 0.019}$
	RMSE	22.504 ± 0.102	0.0174^{*}	22.408 ± 0.013	0.00825^{**}	22.636 ± 0.034	$3.50 \times 10^{-4***}$	$\mathbf{22.264 \pm 0.067}$
KnowAir	MAE	14.114 ± 0.009	0.0001^{***}	14.042 ± 0.023	0.00180^{**}	14.060 ± 0.012	$9.49 \times 10^{-4***}$	$\mathbf{13.966 \pm 0.026}$
	RMSE	21.068 ± 0.093	0.0005^{***}	20.658 ± 0.037	0.1208	20.806 ± 0.009	$3.10 \times 10^{-4***}$	$\mathbf{20.556 \pm 0.052}$
SDWPF	MAE	95.154 ± 0.153	0.0015^{**}	94.726 ± 0.708	0.0251^{*}	96.152 ± 0.856	0.00452^{**}	$\mathbf{92.854 \pm 0.674}$
	RMSE	134.146 ± 1.057	0.0029^{**}	133.648 ± 0.366	0.0465^{*}	135.232 ± 1.358	0.0439^{*}	$\mathbf{131.792 \pm 1.541}$
T-Drive	MAE	16.708 ± 0.104	0.0262^{*}	17.970 ± 0.165	$1.72 \times 10^{-4***}$	18.068 ± 0.188	$1.59 \times 10^{-4***}$	$\mathbf{16.444 \pm 0.130}$
	RMSE	29.662 ± 0.233	0.0152^{*}	31.816 ± 0.319	$1.87 \times 10^{-4***}$	31.812 ± 0.849	$9.31 \times 10^{-4***}$	$\mathbf{29.000 \pm 0.280}$

Across all seven datasets, ERST obtains lower mean MAE and RMSE than the $\psi = 0$ variant, with statistically significant differences in all fourteen comparisons. Removing either $\mathbf{s}_i$ or $\boldsymbol{\mu}_{-i}$ also increases mean MAE on every dataset, and all corresponding differences are statistically significant at the $p < 0.05$ level. For RMSE, ERST obtains a lower mean value in every comparison, with statistically significant differences in all cases except *w/o* $\mathbf{s}_i$ on KnowAir. These results support learning the reference and using both the history of entity i and information from the other entities in its construction.

Analysis of Replacement versus Entity Removal. Because the difference between ERST and *Remove Entity* is relatively small on several datasets, we further evaluate this comparison using five random seeds. This analysis examines the difference between keeping the same entities and replacing only the representation of entity i, and removing entity i, which changes both the set of entities and the normalization. Table 18 reports the mean and standard deviation of MAE and RMSE together with the corresponding p-values.

Across all seven datasets, ERST obtains lower mean MAE and RMSE than *Remove Entity*. The differences are statistically significant at the $p < 0.05$ level in all 14 MAE and RMSE comparisons. Although the differences are small on several datasets, the results across five random seeds provide additional evidence for replacing the representation of one entity while keeping the same entities and normalization, rather than removing the entity.

Table 18: Five-seed comparison between *Remove Entity* and ERST. Results are reported as mean±standard deviation over five random seeds. p-values are obtained from two-sided paired t-tests against ERST using the same random seeds. Lower values are better. $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

Dataset	Metric	Remove Entity	**ERST**	p-value
PEMS03	MAE	14.738 ± 0.264	$\mathbf{14.242 \pm 0.053}$	0.0189^{*}
	RMSE	22.690 ± 0.100	$\mathbf{22.114 \pm 0.087}$	0.00022^{***}
PEMS04	MAE	18.266 ± 0.024	$\mathbf{18.202 \pm 0.029}$	0.0161^{*}
	RMSE	28.274 ± 0.157	$\mathbf{28.026 \pm 0.034}$	0.0394^{*}
PEMS07	MAE	19.744 ± 0.023	$\mathbf{19.450 \pm 0.078}$	0.00117^{**}
	RMSE	32.134 ± 0.088	$\mathbf{31.928 \pm 0.019}$	0.0116^{*}
PEMS08	MAE	14.218 ± 0.040	$\mathbf{14.074 \pm 0.019}$	0.00439^{**}
	RMSE	22.484 ± 0.145	$\mathbf{22.264 \pm 0.067}$	0.0443^{*}
KnowAir	MAE	14.090 ± 0.052	$\mathbf{13.966 \pm 0.026}$	0.0197^{*}
	RMSE	21.082 ± 0.141	$\mathbf{20.556 \pm 0.052}$	0.0030^{**}
SDWPF	MAE	94.568 ± 0.320	$\mathbf{92.854 \pm 0.674}$	0.00958^{**}
	RMSE	135.896 ± 2.685	$\mathbf{131.792 \pm 1.541}$	0.0295^{*}
T-Drive	MAE	17.604 ± 0.364	$\mathbf{16.444 \pm 0.130}$	0.00229^{**}
	RMSE	30.334 ± 0.602	$\mathbf{29.000 \pm 0.280}$	0.00425^{**}

H.2 Sensitivity to the Number of Interaction Layers

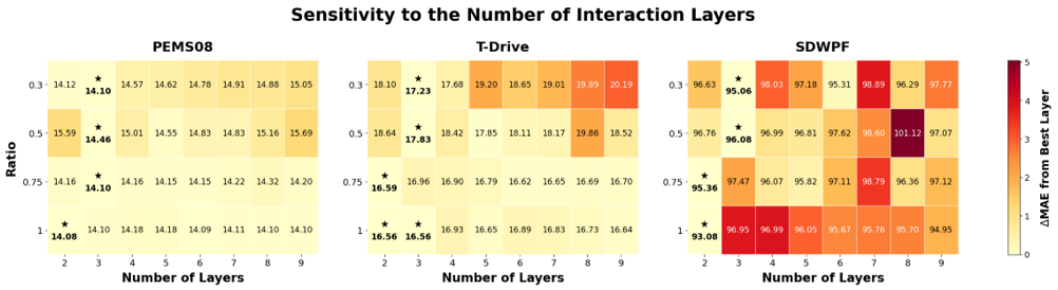


Figure 7: Sensitivity to the number of interaction layers on PEMS08, T-Drive, and SDWPF using different ratios of participating entities. Each cell reports MAE, and the color indicates the increase relative to the lowest MAE within each ratio. Stars denote the lowest value within each ratio. Lower values are better.

Several spatiotemporal forecasting methods use multiple spatial or learned graph interaction layers (Li & Zhu, 2021; Shang et al., 2021). We therefore examine the sensitivity of ERST to the number of interaction layers L on PEMS08, T-Drive, and SDWPF. We vary L from 2 to 9 and repeat the experiment using four different ratios of participating entities. At each update, participating entities are randomly sampled without replacement according to the specified ratio, and the ERST response update is applied only to the sampled entities. Figure 7 reports the resulting MAE.

When all entities participate, $L = 2$ obtains the lowest MAE on PEMS08 and SDWPF and is tied for the lowest MAE on T-Drive. Increasing the number of interaction layers does not consistently reduce the error and increases it in several settings. Similar results are observed when fewer entities participate, where the lowest MAE is generally obtained with two or three interaction layers. Based on these results, we use $L = 2$ as the default setting.

I Additional Analysis of ERST Responses

We further analyze the behavior of the ERST responses during training and at test time. These analyses complement the ablation and response-shuffle experiments reported in the main paper.

Response Assignment across Forecasting Architectures We further examine Response Shuffle across the four forecasting architectures used in Section 4.4. For each architecture, Response Shuffle is applied at inference time to the corresponding ERST model. The response values are kept unchanged, while their assignments across entities are changed. Table 19 reports MAE on the six datasets used in the architecture evaluation.

Table 19: Effect of Response Shuffle across forecasting architectures. Response Shuffle reassigns the responses across entities while preserving their values. Lower MAE is better.

Architecture	Setting	PEMS03	PEMS04	PEMS07	PEMS08	KnowAir	SDWPF
LSTM	Original Architecture	18.67	22.48	30.31	21.02	16.85	105.24
	ERST + Response Shuffle	16.80	21.98	26.93	15.81	14.67	107.68
	ERST	**15.86**	**19.32**	**20.45**	**15.52**	**14.21**	**94.99**
STID	Original Architecture	15.28	19.82	21.61	15.65	**15.04**	102.78
	ERST + Response Shuffle	15.38	18.62	20.00	16.04	15.44	108.65
	ERST	**14.10**	**18.45**	**19.52**	**14.28**	**15.04**	**100.75**
iTransformer	Original Architecture	18.02	24.38	26.40	22.84	15.31	95.12
	ERST + Response Shuffle	17.54	26.17	24.45	19.78	15.28	94.64
	ERST	**16.81**	**22.78**	**23.46**	**17.78**	**15.04**	**94.42**
PatchTST	Original Architecture	21.04	27.40	30.36	21.91	15.79	96.72
	ERST + Response Shuffle	18.80	26.31	29.53	21.10	15.87	96.76
	ERST	**18.62**	**25.31**	**27.09**	**20.07**	**15.25**	**95.16**

Response Shuffle increases MAE relative to ERST in all 24 combinations of forecasting architecture and dataset. This result shows that the performance obtained with ERST across different forecasting architectures depends on using each response for the entity for which it was computed.

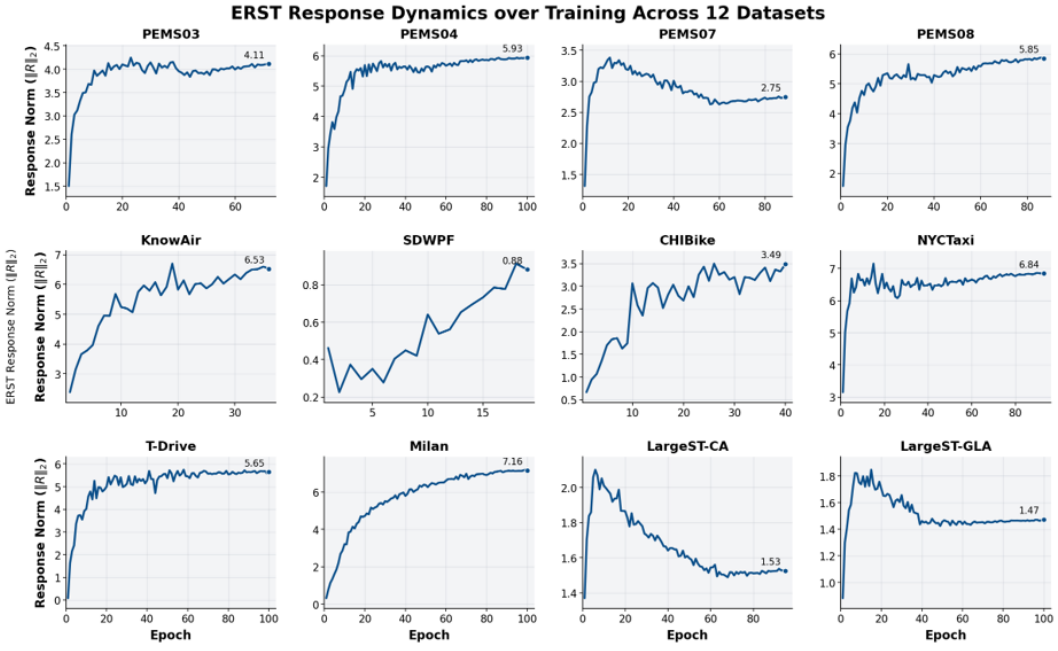


Figure 8: Response norms during training across twelve datasets. The response norms remain nonzero during training and converge to different ranges across datasets. Their absolute magnitudes are not compared directly across datasets.

Response during Training. Figure 8 shows the average response norm during training across all twelve datasets. The responses remain nonzero and converge to different ranges across datasets. The trajectories also differ across datasets, showing that the response magnitude varies with the

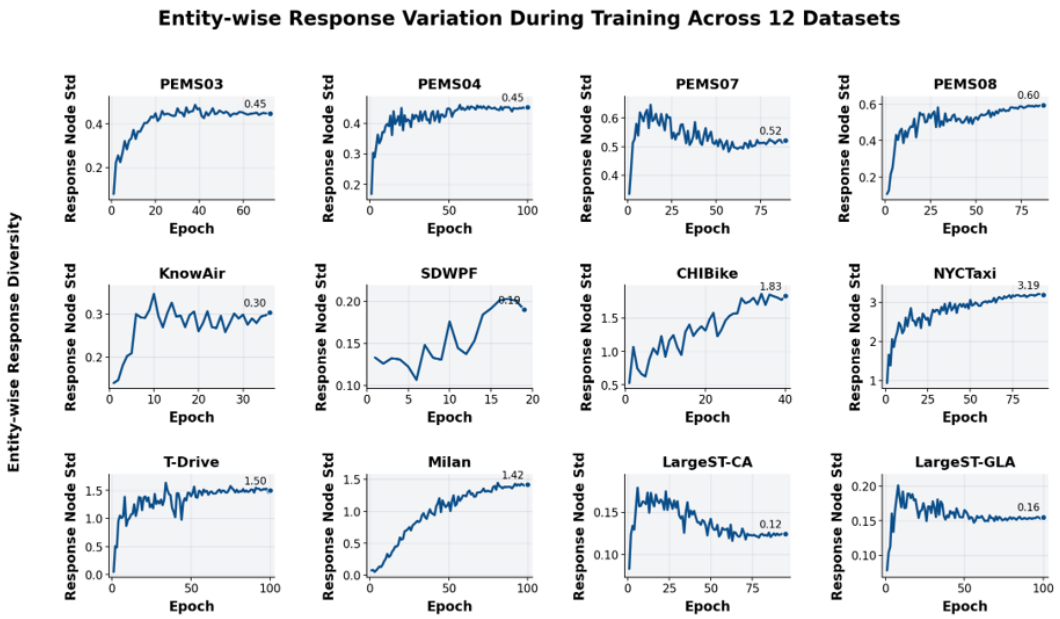


Figure 9: Variation of the ERST responses across entities on twelve datasets. The standard deviation of the response norms across entities is reported. Nonzero values indicate that the response differs across entities rather than taking the same value for all entities.

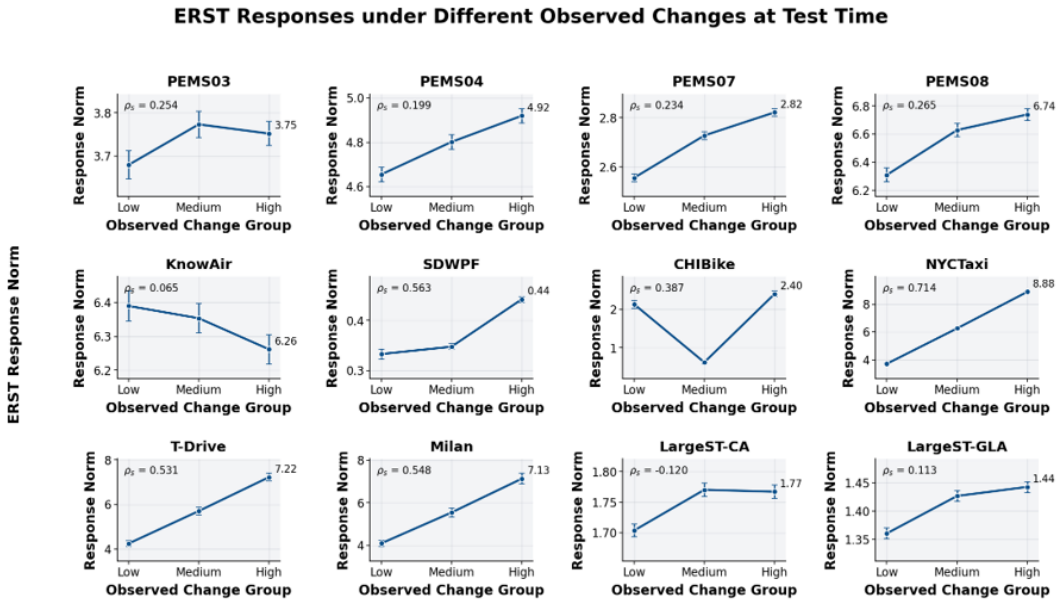


Figure 10: ERST responses at test time for groups with low, medium, and high observed changes across twelve datasets. Error bars denote variability within each group, and ρ_s denotes the Spearman rank correlation between the observed change and response magnitude. The association differs across datasets and is not assumed to be monotonic.

forecasting setting. Because the scales of the representations can differ across datasets, the absolute response magnitudes are not compared directly between datasets.

Variation of Responses across Entities. We next examine whether the responses are identical across entities. Figure 9 reports the standard deviation of the response norms across entities during

training. Nonzero values are observed across all datasets, indicating that the responses differ across entities rather than taking the same value for all entities. This result complements the response-shuffle analysis in the main paper, which examines whether each response should be used for the corresponding entity.

Response under Different Observed Changes. Finally, we examine the response at test time after grouping samples according to the magnitude of the observed change for each entity. Figure 10 shows that the response magnitude differs among the low-, medium-, and high-change groups on most datasets. The association is not consistently monotonic across datasets. This is consistent with the ERST formulation because the response is determined by the difference obtained before and after replacing the representation of one entity, rather than by the observed change magnitude alone. The response can therefore also depend on the representations of the other entities through the function evaluated before and after replacement.

J Efficiency and Scalability

We further evaluate the computational efficiency of ERST on datasets with different numbers of entities. In addition to forecasting accuracy, we report the number of trainable parameters, training time per epoch, and inference time. We also evaluate the consistency of the forecasting results across five random seeds.

Accuracy and Efficiency. We evaluate ERST on PEMS07, SDWPF, T-Drive, and LargeST-CA. Tables 20 and 21 report forecasting accuracy, parameter count, training time, and inference time on PEMS07 and SDWPF. Figure 11 reports MAE and inference time on T-Drive and LargeST-CA, with bubble size indicating the number of trainable parameters. All efficiency measurements use the same experimental environment within each dataset. The measurements are conducted on Google Colab using an NVIDIA A100 GPU.

Table 20: Accuracy and computational efficiency on PEMS07. Par. denotes the number of trainable parameters in millions (M). Train denotes training time per epoch (s), and Infer denotes inference time (s). Lower values are better. Best results are shown in bold and second-best results are underlined.

Model	MAE	RMSE	MAPE	Par. (M)	Train	Infer
MTGNN	21.02	33.89	9.11	<u>0.181</u>	39.33	4.61
STGODE	23.84	36.55	10.10	0.736	157.82	21.82
FEDformer	26.23	40.10	11.75	15.809	70.63	9.50
STID	21.61	32.79	9.23	**0.140**	**7.55**	**1.17**
SOFTS	26.15	41.28	11.01	6.842	35.04	4.23
STPGNN	26.04	39.91	11.45	0.421	74.88	7.47
TWIST	<u>19.70</u>	<u>32.70</u>	8.44	1.121	42.22	3.18
PatchSTG	20.37	33.82	<u>8.11</u>	2.302	22.25	4.37
RAST	21.28	34.21	9.27	0.311	119.28	7.10
FaST	21.22	36.11	8.86	0.485	32.14	3.68
ERST (Ours)	**19.51**	**31.96**	**8.02**	0.256	<u>14.55</u>	<u>1.52</u>

ERST obtains the lowest forecasting errors on PEMS07 and SDWPF while using a relatively small number of parameters and moderate training and inference time. On T-Drive, ERST obtains the lowest MAE with an inference time of 0.20 seconds. On LargeST-CA, which contains 8,600 entities, ERST obtains an MAE of 18.03 with an inference time of 12.90 seconds. Its forecasting error is close to the lowest reported value while its inference time is lower than that of several compared methods. These results show that ERST can be used on datasets with different numbers of entities while maintaining low forecasting error and moderate computational cost.

Statistical Significance. We compare ERST with a corresponding baseline on seven datasets using five random seeds. Table 22 reports the mean and standard deviation of MAE together with the corresponding p-values. Statistical significance is assessed using a two-sided Welch's t-test.

Table 21: Accuracy and computational efficiency on SDWPF. Par. denotes the number of trainable parameters in millions (M). Train denotes training time per epoch (s), and Infer denotes inference time (s). Lower values are better. Best results are shown in bold and second-best results are underlined.

Model	MAE	RMSE	Par. (M)	Train	Infer
MTGNN	<u>94.19</u>	165.75	<u>0.121</u>	17.80	1.52
STGODE	101.19	<u>164.23</u>	0.706	39.88	5.11
FEDformer	107.04	168.07	11.974	21.97	2.16
STID	102.78	178.09	**0.116**	**4.22**	**0.76**
SOFTS	97.14	167.15	6.842	<u>8.83</u>	<u>1.00</u>
STPGNN	94.98	167.64	0.369	23.96	2.47
TWIST	140.47	208.34	0.719	14.80	1.14
PatchSTG	98.66	165.30	2.249	38.17	3.60
RAST	98.56	174.30	0.261	27.75	5.77
FaST	95.32	168.14	0.399	18.01	2.04
ERST (Ours)	**93.08**	**131.20**	0.199	15.66	1.35

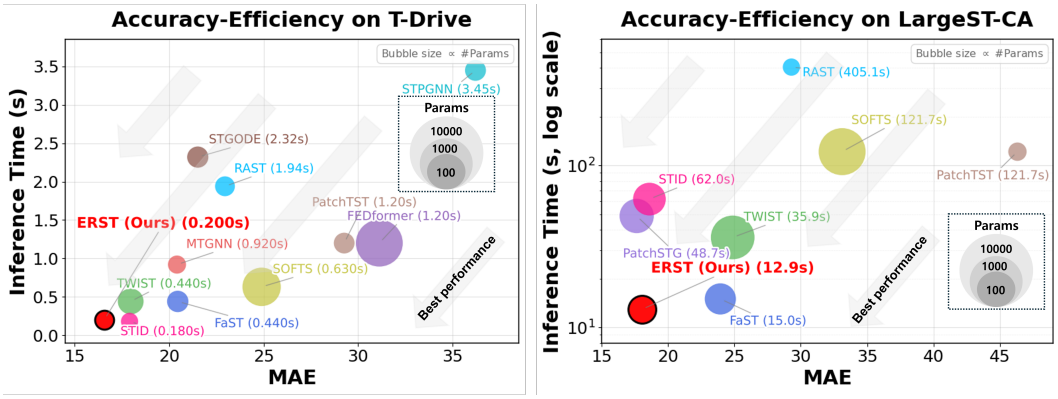


Figure 11: Accuracy–efficiency trade-off on T-Drive and LargeST-CA. The horizontal axis denotes forecasting MAE, the vertical axis denotes inference time, and bubble size is proportional to the number of trainable parameters. Lower-left positions indicate a more favorable accuracy–efficiency trade-off.

Table 22: Five-seed statistical comparison between ERST and a corresponding strong baseline on seven representative datasets. Results are reported as mean±standard deviation. Statistical significance is assessed using a two-sided Welch's t-test. Lower MAE is better. $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

Dataset	Baseline	Baseline MAE	**ERST MAE**	p-value
PEMS03	TWIST	15.083 ± 0.055	$\mathbf{14.242 \pm 0.053}$	$8.06 \times 10^{-9***}$
PEMS04	TWIST	18.567 ± 0.067	$\mathbf{18.202 \pm 0.029}$	$5.81 \times 10^{-5***}$
PEMS07	TWIST	19.760 ± 0.056	$\mathbf{19.450 \pm 0.078}$	$1.47 \times 10^{-4***}$
PEMS08	TWIST	14.227 ± 0.021	$\mathbf{14.074 \pm 0.019}$	$2.22 \times 10^{-6***}$
T-Drive	TWIST	17.903 ± 0.055	$\mathbf{16.444 \pm 0.130}$	$1.34 \times 10^{-6***}$
SDWPF	MTGNN	94.947 ± 1.023	$\mathbf{92.854 \pm 0.674}$	0.00668^{**}
KnowAir	PatchSTG	14.133 ± 0.049	$\mathbf{13.966 \pm 0.026}$	0.000492^{***}

ERST obtains lower mean MAE than the corresponding baseline on all seven datasets, with all differences statistically significant at the $p < 0.01$ level. For PEMS03, PEMS04, PEMS07, PEMS08, T-Drive, and KnowAir, the differences are also significant at the $p < 0.001$ level. These results across five random seeds show that the lower mean MAE is not limited to a single reported run.