When to Trust Memory: Retrieval-Guided Probabilistic Spatiotemporal Forecasting under Distribution Shift

Haochen Lv[1] lvhaochen@bjtu.edu.cn
Jianhao Zhang[1] jianhaozhang@bjtu.edu.cn
Zhichen Lei[1] leizhichen@bjtu.edu.cn
Yang Jiang[1] jiangyang@bjtu.edu.cn
Xiaowei Mao[1] maoxiaowei@bjtu.edu.cn
Shengnan Guo[1,2,*] guoshn@bjtu.edu.cn
Youfang Lin[1,3] yflin@bjtu.edu.cn
Huaiyu Wan[1,3] hywan@bjtu.edu.cn

Abstract

Spatiotemporal forecasting plays an important role in real-world applications, where probabilistic approaches provide uncertainty estimates for decision-making. Real-world systems are inherently non-stationary, leading to distribution shifts between training and test data. However, existing probabilistic methods mainly focus on modeling data distributions under standard settings, while distribution shift methods often handle such shifts in a coarse manner, failing to characterize shift degree and adapt prediction strategy accordingly. As a result, they often produce miscalibrated and unreliable forecasts. To address this issue, we propose REMIND, a retrieval-guided probabilistic framework for spatiotemporal forecasting under distribution shift. REMIND leverages retrieval to estimate how well current observation are supported by historical patterns, enabling adaptive prediction under different shift regimes. It employs spatiotemporal retrieval to construct a memory bank and introduces a dual branch architecture, combining a memory-calibrated probabilistic branch for mild shifts with a deviation-aware probabilistic branch for severe shifts. A shift-adaptive gate, guided by retrieval similarity and memory dispersion, balances the two branches according to shift degree. We further design a hierarchical Gaussian mixture to capture heterogeneous uncertainty over the dual branch predictions, and introduce a memory perturbation training scheme to improve generalization. Experiments on real-world datasets with distribution shifts demonstrate that REMIND consistently outperforms state-of-the-art methods in both forecasting accuracy and probabilistic prediction quality. Code is available at https://github.com/LvHaochenBANG/REMIND.git.

1 Introduction

Spatiotemporal forecasting plays a crucial role in real-world applications, such as transportation [1, 2, 3], hydrology [4], and energy systems [5, 6, 7]. In recent years, probabilistic approaches have gained increasing attention, as they aim to model the underlying distribution of system dynamics [8, 9, 10].

* Corresponding author
[1] School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China.
[2] Key Laboratory of Big Data and Artificial Intelligence in Transportation, Ministry of Education, Beijing, China.
[3] Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, Beijing, China.

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

Beyond point predictions, these methods provide uncertainty quantification, which is essential for risk assessment, decision-making, and early warning [11, 12, 13, 14].

Most existing probabilistic spatiotemporal forecasting methods focus on modeling data distributions under standard settings. However, real-world spatiotemporal systems are inherently non-stationary, leading to distribution shifts, reflected as a mismatch between training and test data distributions [15]. Such shifts commonly arise from evolving system dynamics, including changing patterns, sensor noise, and unexpected events such as accidents or extreme weather [16, 17]. Importantly, distribution shift is inherently continuous rather than binary [18]. At one end, the absence of shift corresponds to in-distribution (ID) scenarios, while at the other end, extreme shifts correspond to out-of-distribution (OOD) scenarios [19]. In practice, many real-world cases lie between these two extremes.

Recently, several methods have been proposed to handle distribution shift. ShifTS [20] improves generalization by learning invariant patterns of input features under different external conditions. TFPS [15] performs adaptive forecasting by dynamically selecting experts specialized in specific patterns. In particular, retrieval-based approaches such as RAFT [21] and STRAP [5] have shown strong potential by leveraging an external memory bank of historical data to retrieve and preserve diverse patterns, especially rare ones, thereby improving predictive performance across diverse distributional conditions.

Despite these advances, handling distribution shift remains challenging in practice. Existing methods often handle distribution shift in a coarse manner, lacking the ability to explicitly characterize shift degree and adapt prediction strategies accordingly, resulting in miscalibrated predictive distributions and unreliable forecasts. Retrieval-based methods, for example, typically retrieve historical samples based on historical observations, rank them by similarity, and reuse their associated future outcomes, implicitly assuming that the retrieved samples can be trusted without assessing their reliability. However, this assumption may fail in two ways. First, under severe distribution shifts, even the most similar historical samples may differ substantially from the current observation, providing limited support for prediction. Second, similar historical observations may correspond to markedly different future outcomes, making input similarity alone insufficient to determine the reliability of retrieved information. Consequently, reusing retrieved samples without assessing their relevance and consistency may bias predictions and undermine uncertainty estimation.

To address these challenges, we propose REMIND, a retrieval-guided probabilistic framework for spatiotemporal forecasting under distribution shift. Unlike conventional approaches that treat retrieval as passive memory lookup, REMIND leverages retrieval as an active signal of how well the current observation is supported by historical patterns. Specifically, it assesses both the relevance of retrieved observations and the consistency of their associated future outcomes to guide adaptive probabilistic forecasting.

Concretely, we design a coarse-to-fine spatiotemporal retrieval mechanism that equips REMIND with a memory bank and achieves both high retrieval quality and efficiency. Based on the retrieval, we construct two complementary branches. A memory-calibrated probabilistic branch predicts from retrieved patterns through lightweight calibration and is effective under mild shifts. A deviation-aware probabilistic branch predicts from the current observation and uses deviations from retrieved patterns as residual corrections, making it more suitable for severe shifts. REMIND then fuses the two branches via a shift-adaptive gate, which leverages retrieval similarity and memory dispersion to estimate shift degree and balance them accordingly. In addition, we design a hierarchical Gaussian mixture to capture heterogeneous uncertainty through inner-branch uncertainty and cross-branch disagreement, and introduce a memory perturbation training scheme to improve generalization under distribution shift.

Our contributions are summarized as follows:

- We propose REMIND, a retrieval-guided probabilistic framework for spatiotemporal forecasting under distribution shift. By treating retrieval as an active signal of support from historical patterns, it adaptively balances a memory-calibrated probabilistic branch and a deviation-aware probabilistic branch according to shift degree, enabling adaptive prediction across different shift regimes.
- We develop a hierarchical Gaussian mixture to model heterogeneous predictive uncertainty over the dual branch predictions, and introduce a memory perturbation training scheme to improve generalization under distribution shift.

- Experiments on real-world datasets from hydrology, traffic and energy domains with spatiotemporal distribution shifts demonstrate that our method consistently outperforms state-of-the-art approaches in both forecasting accuracy and probabilistic prediction quality.

2 Related Work

Probabilistic spatiotemporal forecasting models the predictive distribution of future system states. Existing methods can be broadly divided into parametric distribution models and generative distribution models. Parametric distribution methods assume a predefined distribution family, such as Gaussian distributions, and learn its parameters, as in DeepAR [8], DER [22], SDER [23], and LODE [24]. They are computationally efficient but limited by their distributional assumptions. In contrast, generative distribution models, including diffusion-based methods such as PriSTI [25] and DiffSTG [10] and flow-based methods such as STGNF [9] and STMPN [26], learn complex distributions without fixed forms. While more expressive, they typically rely on repeated sampling, resulting in higher computational cost. These methods mainly focus on modeling data distributions under standard settings, where distribution shift is not explicitly considered. In practice, spatiotemporal systems are often dynamic and non-stationary, and distribution changes may occur over time, which can affect predictive calibration.

To improve generalization under such conditions, several methods have been proposed to mitigate the effects of distribution shift. ShifTS [20] learns invariant representations under varying conditions to enhance generalization across shifting environments, while TFPS [15] dynamically selects specialized experts to adapt to diverse temporal patterns. STONE [27] captures invariant node dependencies through gated spatiotemporal semantic graphs and graph intervention, thereby improving generalization under spatiotemporal distribution shifts. Retrieval-based methods, such as RAFT [21], RATD [28], and STRAP [5], further enhance generalization by leveraging historical memory, allowing models to capture diverse patterns and recall rare or previously observed events. While these approaches have shown promising results, several challenges remain. Specifically, existing methods often do not explicitly characterize the degree of distribution shift and adapt prediction strategy accordingly, resulting in miscalibrated predictive distributions. In addition, these approaches are primarily deterministic in nature and focus on adjusting point estimates rather than modeling the full predictive distribution.

3 Preliminaries

Spatial Graph. We model the spatial structure using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$, where $\mathcal{V}$ is the set of N nodes representing spatial locations, $\mathcal{E}$ is the set of edges capturing the relationships between nodes, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix.

Probabilistic Spatiotemporal Forecasting. At each time step t, let $x_{t,n} \in \mathbb{R}^F$ denote the observation at location n with F variables. Aggregating observations from all locations gives $\boldsymbol{X}_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,N}]^\top \in \mathbb{R}^{N \times F}$, representing multivariate observations over N locations. We use $\boldsymbol{X}_H$ to denote the historical sequence over the past T time steps, corresponding to $\boldsymbol{X}_{t-T+1:t} \in \mathbb{R}^{T \times N \times F}$, and use $\boldsymbol{X}_P$ to denote the future sequence over the next T time steps, corresponding to $\boldsymbol{X}_{t+1:t+T} \in \mathbb{R}^{T \times N \times F}$. Given $\boldsymbol{X}_H$, the goal is to learn a probabilistic mapping $\mathcal{F}$ that characterizes the distribution of $\boldsymbol{X}_P$ and produces its predictive statistics, i.e., $(\boldsymbol{\mu}_P, \boldsymbol{\sigma}_P) = \mathcal{F}(\boldsymbol{X}_H, \mathcal{G})$, where $\boldsymbol{\mu}_P, \boldsymbol{\sigma}_P \in \mathbb{R}^{T \times N \times F}$.

4 REMIND

4.1 Overview

The key idea of REMIND is to treat retrieval similarity not only as auxiliary evidence for prediction, but also as a signal indicating how well the current observation is supported by historical patterns. REMIND assesses both the relevance of retrieved observations and the consistency of their associated future outcomes, enabling adaptive memory utilization under distribution shift. Intuitively, strong historical support suggests that retrieved patterns are both relevant and informative, allowing their associated targets to be reused. In contrast, weak historical support may arise from substantial deviations from historical observations or inconsistent future outcomes among similar patterns. In such cases, the model reduces its reliance on memory, predicts primarily from the current observation, and uses deviations from retrieved patterns as residual corrections.

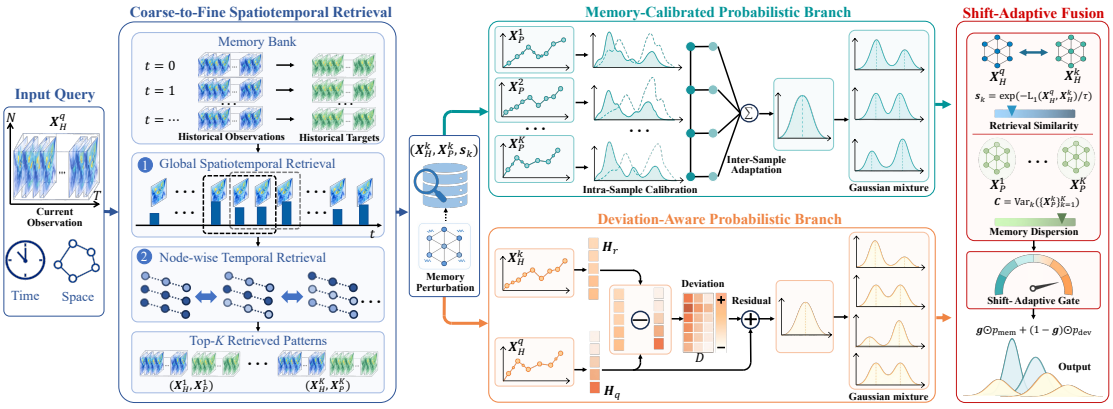

Figure 1: The overall architecture of REMIND. REMIND performs coarse-to-fine spatiotemporal retrieval to construct a memory bank and employs a dual branch architecture for probabilistic forecasting under distribution shift. A shift-adaptive gate dynamically balances the branches based on retrieval similarity and memory dispersion, and a hierarchical Gaussian mixture models heterogeneous predictive distribution.

As shown in Fig.1, REMIND constructs a memory bank through coarse-to-fine spatiotemporal retrieval and employs a dual branch architecture consisting of a memory-calibrated probabilistic branch and a deviation-aware probabilistic branch. The memory-calibrated probabilistic branch performs lightweight calibration on retrieved targets, while the deviation-aware probabilistic branch models residual deviations from retrieved patterns. A shift-adaptive gate balances the two branches according to retrieval similarity and memory dispersion. Finally, REMIND outputs a hierarchical Gaussian mixture distribution to capture heterogeneous uncertainty under different shift regimes.

4.2 Coarse-to-Fine Spatiotemporal Retrieval

Let $\mathcal{D} = \{(\boldsymbol{X}_H^i, \boldsymbol{X}_P^i)\}_{i=1}^R$ denote a memory bank of R historical observation-target pairs. In an ideal spatiotemporal retrieval mechanism, given a query observation $\boldsymbol{X}_H^q$, the system aims to rapidly and efficiently retrieve the top K most similar samples $\mathcal{D}_{\text{ret}} = \{\boldsymbol{X}_H^k, \boldsymbol{X}_P^k, \boldsymbol{s}_k\}_{k=1}^K$, where $\boldsymbol{s}_k$ denotes the similarity score between $\boldsymbol{X}_H^q$ and $\boldsymbol{X}_H^k$.

Retrieving spatiotemporal data is non-trivial due to the entanglement of spatial and temporal dimensions. Accordingly, spatiotemporal retrieval can be formulated at four levels: global spatiotemporal matching, node-wise temporal matching, time-wise spatial matching, and point-wise matching. Global spatiotemporal matching preserves overall structure with high efficiency but enforces strict temporal alignment across nodes. In contrast, node-wise temporal matching retrieves patterns independently for each node based on its temporal dynamics, enabling more flexible and fine-grained retrieval, but requiring independent matching for each node and thus increasing computational cost. Meanwhile, time-wise spatial matching disrupts temporal continuity, and point-wise matching ignores global spatiotemporal structure, limiting their ability to capture coherent spatiotemporal patterns (see App. B for further discussion).

To balance retrieval quality and efficiency, we propose a coarse-to-fine spatiotemporal retrieval framework consisting of global spatiotemporal retrieval followed by node-wise temporal retrieval.

In the first stage, we perform global spatiotemporal retrieval over $\mathcal{D}$, where samples are constructed via sliding windows with a stride of 1 along the temporal dimension. Treating the current observation $\boldsymbol{X}_H^q \in \mathbb{R}^{T\times N\times F}$ as a whole, we select a candidate set $\mathcal{C} = \{(\boldsymbol{X}_H^k, \boldsymbol{X}_P^k, \boldsymbol{s}_k)\}_{k=1}^{\tilde{K}}$ based on global spatiotemporal similarity. This stage narrows the search space to historically relevant samples with similar overall spatiotemporal patterns.

In the second stage, we perform node-wise temporal retrieval within the candidate set $\mathcal{C}$. Specifically, for each query node n, we compare its temporal pattern $\boldsymbol{X}_{H,n}^q$ with the candidate nodes drawn from a node set $\mathcal{N}(n)$, which includes both physically connected neighbors and functionally similar nodes defined by Pearson correlation. The cross-spatial similarity can be written as

$$s_{k,n,m} = \text{Sim}\left(\boldsymbol{X}_{H,n}^q, \boldsymbol{X}_{H,m}^k\right), \quad m \in \mathcal{N}(n). \tag{1}$$

Based on these similarities, the refined retrieval results can be expressed as $\mathcal{D}_{\text{ret}}^{(n)} \subseteq \{\boldsymbol{X}_{H,m}^k, \boldsymbol{X}_{P,m}^k, s_{k,n,m}\}_{k=1}^K$. After processing all nodes, we obtain the final retrieval set $\mathcal{D}_{\text{ret}} = \{\boldsymbol{X}_H^k, \boldsymbol{X}_P^k, \boldsymbol{s}_k\}_{k=1}^K$. Compared with global retrieval, this stage allows the model to capture local temporal misalignment commonly observed under distribution shift. It also leverages similar dynamics from correlated neighboring nodes, thereby enriching retrieval diversity.

The similarity score for the k-th retrieved sample is defined as

$$\boldsymbol{s}_k = \exp\left(-\mathrm{L}_1(\boldsymbol{X}_H^q, \boldsymbol{X}_H^k)/\tau\right), \tag{2}$$

where $\mathrm{L}_1(\cdot,\cdot)$ denotes the ℓ_1 distance and τ is a temperature parameter that controls the concentration of the similarity scores [5]. This formulation captures overall similarity and supports fully parallel computation of similarities between $\boldsymbol{X}_H^q$ and the entire memory bank.

4.3 Memory-Calibrated Probabilistic Branch

The memory-calibrated probabilistic branch is designed for scenarios where retrieved patterns remain reliable under mild distribution shifts. In such cases, the current observation is still well supported by historical patterns, allowing retrieved targets to provide informative future references. Therefore, instead of predicting future dynamics from scratch, this branch directly leverages retrieved targets and performs lightweight query-conditioned calibration to adapt them to current observations.

We first encode the current observation and the k-th retrieved observation:

$$\boldsymbol{H}_q = E_q(\boldsymbol{X}_H^q), \qquad \boldsymbol{H}_r^k = E_r(\boldsymbol{X}_H^k), \tag{3}$$

where $\boldsymbol{H}_q, \boldsymbol{H}_r^k \in \mathbb{R}^{N\times T\times d}$. E_q and E_r are lightweight spatiotemporal encoders composed of GCNs [29] and temporal attention [30].

Next, for each retrieved target $\boldsymbol{X}_P^k$, we apply a lightweight calibration:

$$\tilde{\boldsymbol{X}}_P^k = \boldsymbol{a}_k \odot \boldsymbol{X}_P^k + \boldsymbol{b}_k, \tag{4}$$

where $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$ are learnable parameters, and $\odot$ denotes element-wise multiplication. To improve stability and prevent overfitting, the parameters are modeled in a low-rank separable form [31]. Specifically, $\boldsymbol{a}_k$ is constructed as the outer product of a spatial factor $\boldsymbol{a}_k^{\text{node}} \in \mathbb{R}^N$ and a temporal factor $\boldsymbol{a}_k^{\text{time}} \in \mathbb{R}^T$, i.e., $\boldsymbol{a}_k = \boldsymbol{a}_k^{\text{node}}(\boldsymbol{a}_k^{\text{time}})^\top$, with $\boldsymbol{b}_k$ defined analogously.

In addition to the above intra-sample calibration, we further perform inter-sample adaptation. We compute a relevance weight for each retrieved sample:

$$\boldsymbol{\alpha}_k = \mathrm{softmax}\Big(\boldsymbol{s}_k + d^{-1/2}\,\langle \bar{\boldsymbol{H}}_q, \bar{\boldsymbol{H}}_r^k\rangle\Big), \tag{5}$$

where $\bar{\boldsymbol{H}}_q, \bar{\boldsymbol{H}}_r^k \in \mathbb{R}^{N\times d}$ are temporally aggregated representations obtained via temporal convolution [32] followed by pooling. The Frobenius inner product $\langle\cdot,\cdot\rangle$ quantifies the similarity between the query and retrieved representations, while $\boldsymbol{s}_k$ provides a retrieval-based prior. Unlike directly normalizing retrieval similarity, the query-conditioned weighting is not solely determined by historical similarity, allowing the model to adaptively adjust the contributions of retrieved samples. Using these relevance weights, the base predictive mean of memory-calibrated probabilistic branch is computed as a weighted aggregation of calibrated targets $\boldsymbol{\mu}_{\text{mem}} = \sum_{k=1}^K \boldsymbol{\alpha}_k \tilde{\boldsymbol{X}}_P^k$.

To capture uncertainty, we relate the predictive variance to both representation features and memory dispersion $\boldsymbol{C} = \mathrm{Var}_k(\{\boldsymbol{X}_P^k\}_{k=1}^K) \in \mathbb{R}^{T\times N\times F}$. Intuitively, larger dispersion indicates greater variability among retrieved targets, implying higher uncertainty over possible future outcomes. Based on this, the base variance is estimated as:

$$\boldsymbol{\sigma}_{\text{mem}} = \mathrm{Softplus}\big(f_\sigma(\boldsymbol{H}_q, \boldsymbol{H}_r) + f_c(\boldsymbol{C})\big), \tag{6}$$

where $\boldsymbol{H}_r = \sum_{k=1}^K \boldsymbol{\alpha}_k \boldsymbol{H}_r^k$ denotes the aggregated retrieved representation, f_σ and f_c are implemented as lightweight MLPs.

To better capture the inherent heterogeneity of spatiotemporal forecasting, we parameterize the predictive distribution of the memory-calibrated probabilistic branch as a Gaussian mixture [33]:

$$p_{\text{mem}}(\boldsymbol{X}_P \mid \boldsymbol{X}_H) = \sum_{i=1}^{I_{\text{mem}}} \boldsymbol{\pi}_i\, \mathcal{N}\Big(\boldsymbol{X}_P; \boldsymbol{\mu}_{\text{mem}} + \Delta\boldsymbol{\mu}_i, (\boldsymbol{\sigma}_{\text{mem}} + \Delta\boldsymbol{\sigma}_i)^2\Big), \tag{7}$$

where the base terms $\boldsymbol{\mu}_{\text{mem}}$ and $\boldsymbol{\sigma}_{\text{mem}}$ provide a retrieval-based prior, while the residual terms $\Delta\boldsymbol{\mu}_i$ and $\Delta\boldsymbol{\sigma}_i$ enable flexible refinement for each component. The mixture weights $\boldsymbol{\pi}_i$ and the residual parameters $\Delta\boldsymbol{\mu}_i, \Delta\boldsymbol{\sigma}_i$ are predicted from the memory context $(\boldsymbol{H}_q, \boldsymbol{H}_r)$. In addition to providing expressive distribution modeling, the mixture formulation also facilitates the subsequent probabilistic fusion between branches, since the gate-weighted combination of branch-wise mixtures naturally forms a unified hierarchical Gaussian mixture distribution.

4.4 Deviation-Aware Probabilistic Branch

The deviation-aware probabilistic branch is designed for scenarios where retrieved patterns become less reliable under severe distribution shifts. In such cases, retrieved targets may no longer accurately represent future evolution, while deviations between current observations and historical patterns become informative for characterizing shift dynamics. Therefore, instead of directly reusing retrieved targets, this branch predicts from current observations and leverages deviations from retrieved patterns as residual corrections.

Similar to the memory-calibrated probabilistic branch, we encode the current observation and retrieved observations into spatiotemporal representations $\boldsymbol{H}_q$ and $\{\boldsymbol{H}_r^k\}_{k=1}^K$, via an encoder which is independent of that in the memory-calibrated probabilistic branch. The retrieved representations are then aggregated by similarity weights as $\boldsymbol{H}_r = \sum_{k=1}^K \boldsymbol{s}_k \boldsymbol{H}_r^k$, yielding a reference representation that summarizes historical patterns relevant to the current observation.

To quantify the deviation between the current observation and the retrieved reference representation, we construct a deviation representation based on their feature differences and interactions. Specifically, we define

$$\boldsymbol{D} = \phi([\boldsymbol{H}_q - \boldsymbol{H}_r,\ \boldsymbol{H}_q \odot \boldsymbol{H}_r]), \tag{8}$$

where $[\cdot,\cdot]$ denotes concatenation, and $\phi(\cdot)$ is implemented as MLPs that maps the combined features to a compact deviation representation. Based on $\boldsymbol{D}$, the base predictive mean and variance of the deviation-aware probabilistic branch are defined as

$$\boldsymbol{\mu}_{\text{dev}} = f_\mu(\boldsymbol{H}_q) + f_\mu^d(\boldsymbol{D}), \quad \boldsymbol{\sigma}_{\text{dev}} = \text{Softplus}(f_\sigma(\boldsymbol{H}_q)) + f_\sigma^d(\boldsymbol{D}), \tag{9}$$

where $f_\mu(\boldsymbol{H}_q)$ and $f_\sigma(\boldsymbol{H}_q)$ model the primary prediction from the current observation, while $f_\mu^d(\boldsymbol{D})$ and $f_\sigma^d(\boldsymbol{D})$ provides deviation-aware adjustments. This residual design enables the model to adapt its prediction and uncertainty according to deviations from historical patterns.

Finally, similar to the memory-calibrated probabilistic branch, the deviation-aware probabilistic branch also outputs a Gaussian mixture distribution:

$$p_{\text{dev}}(\boldsymbol{X}_P \mid \boldsymbol{X}_H) = \sum_{i=1}^{I_{\text{dev}}} \boldsymbol{\pi}_i\, \mathcal{N}\Big(\boldsymbol{X}_P; \boldsymbol{\mu}_{\text{dev}} + \Delta\boldsymbol{\mu}_i, (\boldsymbol{\sigma}_{\text{dev}} + \Delta\boldsymbol{\sigma}_i)^2\Big). \tag{10}$$

Compared with the memory-calibrated probabilistic branch, the deviation-aware probabilistic branch requires more mixture components because it cannot directly rely on retrieved targets to provide structured future patterns. Under severe distribution shift, the model must infer possible future evolutions from deviations between current observations and historical patterns, which generally introduces greater ambiguity in future evolution. Additional mixture components therefore improve the capacity of the branch to model heterogeneous uncertainties and diverse future outcomes.

4.5 Shift-Adaptive Probabilistic Fusion

Since the two probabilistic branches of REMIND are designed for different shift regimes, their contributions should adapt according to the degree of distribution shift. To achieve this, we introduce a shift-adaptive probabilistic fusion module that dynamically balances the two branches based on retrieval similarity $\boldsymbol{s}$ and memory dispersion $\boldsymbol{C}$. These signals provide complementary indicators of distribution shift and guide adaptive probabilistic fusion under varying shift conditions.

Intuitively, retrieval similarity $\boldsymbol{s}$ reflects how well the current observation is supported by historical patterns. Higher similarity generally indicates that the current observation remains closer to the training distribution, making retrieved targets more reliable for prediction. In contrast, memory dispersion $\boldsymbol{C}$ characterizes the variability among retrieved targets. Even when retrieved observations are historically similar, their corresponding targets may still exhibit diverse future evolutions,

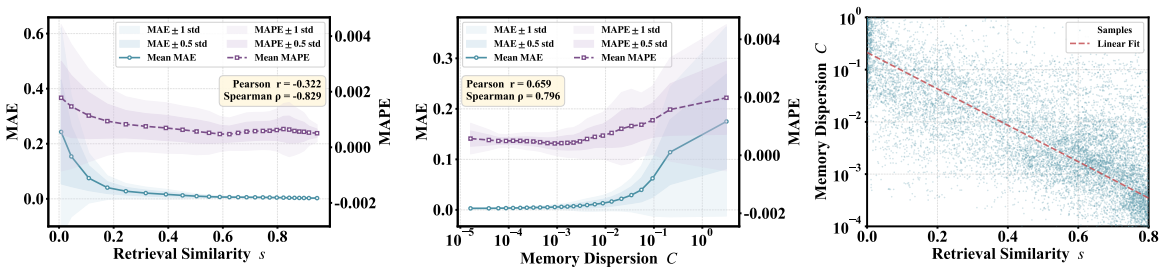

Figure 2: Empirical analysis of retrieval similarity and memory dispersion under distribution shift on the CAMELS dataset. We analyze their relationships with error (MAE/MAPE) between retrieved targets and ground-truth future targets, error variance, and their pairwise correlation.

resulting in larger dispersion. This suggests weaker consensus in historical support and potentially greater uncertainty under distribution shift. Therefore, $\boldsymbol{s}$ and $\boldsymbol{C}$ provide complementary signals for characterizing distribution shift and guiding adaptive probabilistic fusion.

From a statistical perspective, the historical observation-target pairs in the memory bank can be viewed as samples from a joint distribution, which can be factorized as $p(\boldsymbol{X}_H, \boldsymbol{X}_P) = p(\boldsymbol{X}_H)p(\boldsymbol{X}_P \mid \boldsymbol{X}_H)$. Retrieval similarity $\boldsymbol{s}$ can indicate how well the current observation aligns with historical samples from the marginal distribution $p(\boldsymbol{X}_H)$, while memory dispersion $\boldsymbol{C}$ can reflect the variability of the conditional distribution $p(\boldsymbol{X}_P \mid \boldsymbol{X}_H)$ in the neighborhood of the current observation. These two statistics thus provide complementary evidence for guiding adaptive probabilistic fusion.

Empirical analysis further confirms the effectiveness of retrieval similarity $\boldsymbol{s}$ and memory dispersion $\boldsymbol{C}$ as indicators of distribution shift, as shown in Fig. 2. Overall, lower similarity and larger dispersion are associated with larger forecasting errors and larger error variance, indicating weaker historical support under distribution shift. Moreover, the relationship between the two signals remains relatively dispersed, suggesting that they capture different aspects of distribution shift. These observations support the use of both $\boldsymbol{s}$ and $\boldsymbol{C}$ for adaptive probabilistic fusion.

Motivated by the above observations, we compute the shift-adaptive gate:

$$\boldsymbol{g} = \mathrm{Sigmoid}\left(f_g([\bar{\boldsymbol{s}}, \boldsymbol{s}_1, \boldsymbol{C}, \boldsymbol{D}])\right), \tag{11}$$

where $\bar{\boldsymbol{s}}$ and $\boldsymbol{s}_1$ denote the average and top-1 retrieval similarities, respectively. $\boldsymbol{C}$ is defined in Sec.4.3, while deviation $\boldsymbol{D}$ is defined by Eq.8. Here, $\boldsymbol{D}$ complements $\boldsymbol{s}$ by modeling deviation patterns between current observations and historical patterns.

To guide the shift-adaptive gate toward shift-consistent behavior, we impose soft regularization using $\boldsymbol{s}$ and $\boldsymbol{C}$. First, we encourage the gate $\boldsymbol{g}$ to increase with higher retrieval similarity $\boldsymbol{s}$:

$$\mathcal{L}_{\mathrm{sim}} = -\mathbb{E}\left[\boldsymbol{s}\log\boldsymbol{g} + (1-\boldsymbol{s})\log(1-\boldsymbol{g})\right]. \tag{12}$$

For memory dispersion $\boldsymbol{C}$, we introduce a complementary regularization term $\mathcal{L}_{\mathrm{dis}}$, which shares the same form as $\mathcal{L}_{\mathrm{sim}}$ and encourages lower gate values under larger dispersion. These regularizations provide soft directional guidance for gate optimization rather than enforcing hard constraints.

Based on the learned shift-adaptive gate $\boldsymbol{g}$, the final predictive distribution is obtained via a weighted combination of the branch-wise Gaussian mixtures defined in Eqs.7 and10:

$$p(\boldsymbol{X}_P|\boldsymbol{X}_H) = \boldsymbol{g} \odot p_{\mathrm{mem}}(\boldsymbol{X}_P|\boldsymbol{X}_H) + (1-\boldsymbol{g}) \odot p_{\mathrm{dev}}(\boldsymbol{X}_P|\boldsymbol{X}_H). \tag{13}$$

This formulation establishes a hierarchical Gaussian mixture, enabling REMIND to jointly model intra-branch multimodality and cross-branch adaptive fusion. From a probabilistic perspective, the resulting predictive variance can be decomposed as

$$\mathrm{Var}[\boldsymbol{X}_P] = \boldsymbol{g}\,\bar{\boldsymbol{\sigma}}^2_{\mathrm{mem}} + (1-\boldsymbol{g})\,\bar{\boldsymbol{\sigma}}^2_{\mathrm{dev}} + \boldsymbol{g}(1-\boldsymbol{g})\left(\bar{\boldsymbol{\mu}}_{\mathrm{mem}} - \bar{\boldsymbol{\mu}}_{\mathrm{dev}}\right)^2, \tag{14}$$

where $\bar{(\cdot)}$ denotes the branch-wise aggregated means and variances of the corresponding Gaussian mixtures. This formulation shows that the final predictive uncertainty depends on both inner-branch uncertainty, characterized by the first two terms, and cross-branch disagreement, captured by the last term. Detailed derivation is provided in App. A.

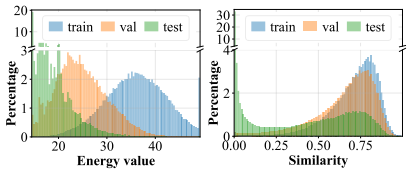

Figure 3: Distribution on ENERGY.

Table 4: Comparison with distribution shift methods.

Metric	ShiftTS	STONE	TFPS	STRAP	**REMIND**
MAE	9.51±0.18	8.85±0.16	9.16±0.17	8.95±0.15	**8.73**±0.14
RMSE	34.67±0.62	33.81±0.58	34.35±0.60	34.33±0.57	**32.26**±0.55
MAPE	27.24±0.85	28.05±0.92	27.23±0.88	29.85±1.05	**25.30**±0.80

4.6 Training Objective and Memory Perturbation

The model is trained using the negative log-likelihood (NLL) of the predictive distribution:

$$\mathcal{L}_{\text{NLL}} = -\mathbb{E}\left[\log p(\boldsymbol{X}_P \mid \boldsymbol{X}_H)\right]. \tag{15}$$

We further derive the explicit form of the negative log-likelihood under the hierarchical Gaussian mixture by expanding the two-level mixture into a unified mixture (see App. A).

Following the gate regularization terms introduced in Eq. 12, the overall objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \lambda(\mathcal{L}_{\text{sim}} + \mathcal{L}_{\text{dis}}). \tag{16}$$

To further improve generalization under distribution shift, we adopt a memory perturbation training scheme to prevent the model from over-relying on highly confident retrievals. Specifically, we randomly attenuate the memory gate as $\tilde{\boldsymbol{g}} = \rho \cdot \boldsymbol{g}$ with $\rho \sim \text{Uniform}(1-p, 1)$, which serves as a soft memory dropout. Here, $p \in [0, 1]$ controls the degree of attenuation. In addition, we perturb the retrieved observations and targets as

$$\tilde{\boldsymbol{X}}_H^k = \boldsymbol{X}_H^k + (1-\rho)\,\boldsymbol{\epsilon}_H, \quad \tilde{\boldsymbol{X}}_P^k = \boldsymbol{X}_P^k + (1-\rho)\,\boldsymbol{\epsilon}_P, \tag{17}$$

where $\boldsymbol{\epsilon}_H, \boldsymbol{\epsilon}_P \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$ denote independent Gaussian noise. These perturbations simulate imperfect retrieval and weak-support scenarios that are more likely to occur under distribution shift, where retrieval similarity is generally lower than in training data. By exposing the model to degraded memory conditions during training, REMIND is encouraged to reduce over-reliance on highly confident retrievals and improve generalization under challenging shift conditions.

5 Experiments

5.1 Experimental Settings

Datasets and Metrics. We conduct experiments on three real-world spatiotemporal datasets, namely CAMELS [34], PEMS [35], and ENERGY [36], which cover hydrology, transportation, and energy domains. These datasets span diverse temporal and spatial resolutions and cover long observation periods, leading to pronounced distribution shifts, as shown in Fig. 3. We evaluate REMIND using both deterministic forecasting metrics, including MAE, RMSE, and MAPE, and probabilistic metrics, including CRPS [37] and the Energy Score (ES) [9]. (Details in App. C.1 and C.2)

Baselines. We compare REMIND with eight advanced probabilistic forecasting methods across the two categories defined above. The parametric distribution models include DeepAR [8], LODE [24], DER [22], and SDER [23]. The generative distribution models include PriSTI [25], DiffSTG [10], STGNF [9], and STMPN [26]. In addition, we compare REMIND with distribution shift methods, including ShifTS [20], STONE [27], TFPS [15], and STRAP [5]. As these methods are deterministic and do not model predictive uncertainty, the comparison is conducted using deterministic metrics.

Implementation Details. All datasets are split into training, validation, and test sets with a ratio of 6:2:2, and Z-score normalization is applied using training statistics. The model predicts the next 12 time steps based on the previous 12 observations. We adopt the Adam optimizer [38] with an initial learning rate of $1\text{e}{-4}$. For fair comparison, baseline hyperparameters are carefully tuned around their recommended or default settings. All results are averaged over three runs with different random seeds. Additional implementation details settings are provided in App. C.3.

5.2 Performance Comparison

As shown in Tab. 1, REMIND achieves the best performance across all datasets and evaluation metrics. Compared with conventional parametric models, REMIND improves distributional expressiveness through the hierarchical Gaussian mixture while retaining the efficiency of likelihood-based prediction.

Table 1: Overall performance comparison. The best is **bold** and the second best is underlined.

Dataset	Metric	DeepAR	LODE	DER	SDER	PriSTI	DiffSTG	STGNF	STMPN	**REMIND**
CAMELS	MAE	9.85	10.52	9.68	9.32	9.05	8.93	8.96	9.23	**8.73**
	RMSE	38.95	34.87	38.73	37.95	36.84	34.12	35.93	35.93	**32.26**
	MAPE	29.53	45.14	37.85	38.24	35.65	30.96	29.47	34.25	**25.30**
	CRPS	0.3611	0.3252	0.4210	0.3056	0.2955	0.3028	0.2818	0.2577	**0.2232**
	ES	28.73	31.27	36.99	27.67	28.18	27.46	26.46	29.86	**26.04**
PEMS	MAE	19.82	21.45	20.25	19.64	19.38	19.13	18.55	17.91	**16.63**
	RMSE	32.95	34.82	33.42	32.64	32.10	31.75	30.52	29.28	**26.88**
	MAPE	20.85	23.93	22.10	21.66	20.31	20.98	19.46	18.60	**17.76**
	CRPS	0.0942	0.0985	0.1013	0.0915	0.0906	0.0898	0.0887	0.0868	**0.0854**
	ES	520.75	560.48	548.93	505.37	502.85	498.69	472.92	486.53	**453.21**
ENERGY	MAE	2.12	2.45	2.25	2.18	2.05	2.08	1.93	2.30	**1.72**
	RMSE	3.95	3.62	3.98	3.85	3.23	3.88	2.95	4.77	**2.39**
	MAPE	24.84	28.97	27.60	26.95	25.41	24.74	23.17	25.51	**21.49**
	CRPS	0.1628	0.1452	0.1745	0.1397	0.1345	0.1218	0.1372	0.1437	**0.1027**
	ES	19.66	20.85	21.48	19.12	18.97	18.10	19.23	20.72	**16.92**

Table 2: Ablation study on core architecture.

Variant	MAE	RMSE	MAPE	CRPS	ES
w/o Mem	2.90	4.03	46.29	0.1664	27.25
w/o Dev	3.12	4.14	46.75	0.1935	30.43
w/o Gate	3.01	4.09	48.48	0.1932	30.67
REMIND	**1.72**	**2.39**	**21.49**	**0.1027**	**16.92**

Table 3: Ablation study on key mechanisms.

Variant	MAE	RMSE	MAPE	CRPS	ES
w/o MD	2.06	2.85	30.38	0.1268	19.63
w/o MP	1.78	2.47	23.28	0.1050	17.29
w/o GM	1.74	2.41	21.73	0.1173	19.57
REMIND	**1.72**	**2.39**	**21.49**	**0.1027**	**16.92**

Although generative methods provide stronger distribution modeling capacity, they do not estimate historical support. In contrast, REMIND leverages retrieval as shift signals to adaptively balance memory reuse and deviation-aware prediction, leading to better generalization under distribution shift. We further compare REMIND with representative distribution shift methods in Tab. 4. Although these baselines mainly focus on deterministic prediction, REMIND still achieves the best performance on CAMELS. The results demonstrate that characterizing shift degree and adaptively adjusting prediction strategies are more effective than a fixed adaptation strategy to all samples.

5.3 Ablation Study

Tabs.2 and3 report two groups of ablation studies on ENERGY. Removing the memory-calibrated probabilistic branch (Mem), deviation-aware probabilistic branch (Dev), or shift-adaptive gate (Gate) all leads to clear performance degradation, confirming that REMIND relies on the cooperation of retrieval reuse, deviation-aware prediction, and adaptive fusion. Among these components, removing the Dev causes the largest degradation, highlighting the importance of modeling deviations from historical patterns under strong distribution shift. The gate is also critical, showing that continuously varying shift conditions cannot be handled by static fusion. We further analyze several key mechanisms, including memory dispersion (MD), memory perturbation (MP), and Gaussian mixture modeling (GM). All three contribute to the final performance. In particular, removing memory dispersion causes the largest drop, supporting its role as an indicator of retrieval reliability under distribution shift. Memory perturbation improves robustness by reducing over-reliance on highly confident retrieval during training, while Gaussian mixture modeling improves probabilistic quality by better capturing diverse future modes.

5.4 Hyperparameter Study

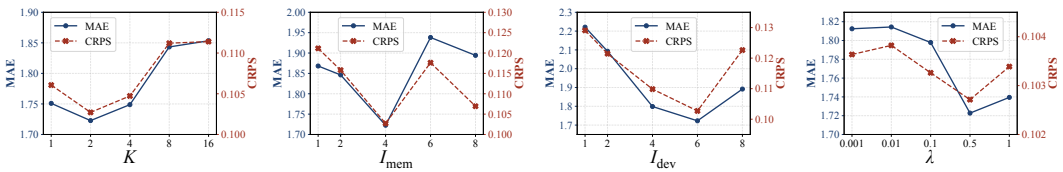

Figure 4: Hyperparameter sensitivity on ENERGY.

Fig. 4 presents the hyperparameter analysis of REMIND. The results for I_{mem} and I_{dev} are consistent with the design of the hierarchical Gaussian mixture: the memory-calibrated probabilistic branch requires fewer components because retrieved targets already provide structured future patterns, while

Table 5: Memory dispersion analysis on ENERGY. We select the top third of test samples by retrieval similarity and divide them into three groups based on the tertiles of C.

Dispersion	Memory branch MAE / CRPS	Deviation branch MAE / CRPS	Mean g	REMIND MAE / CRPS
Low	1.38 / 0.0795	1.52 / 0.0860	0.91	1.35 / 0.0780
Medium	1.60 / 0.0925	1.57 / 0.0910	0.66	1.52 / 0.0875
High	1.92 / 0.1110	1.83 / 0.1045	0.38	1.78 / 0.0994

the deviation-aware probabilistic branch benefits from more components to model diverse future evolutions under severe distribution shift. In addition, the hyperparameter analysis of λ indicates that the gate benefits from soft shift-guided regularization without being overly constrained.

5.5 Gate Behavior Analysis

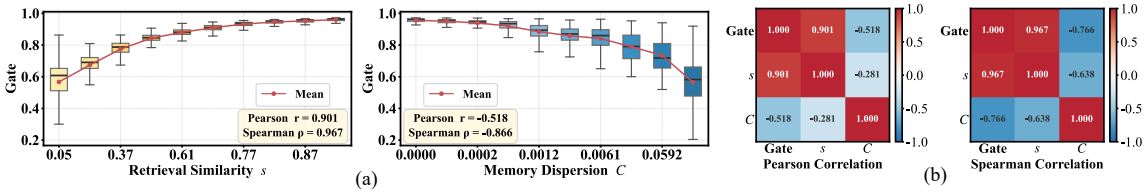

Figure 5: (a) Box plots of learned gate values across retrieval similarity and memory dispersion levels. (b) Pearson and Spearman correlations among learned gate values and these two signals.

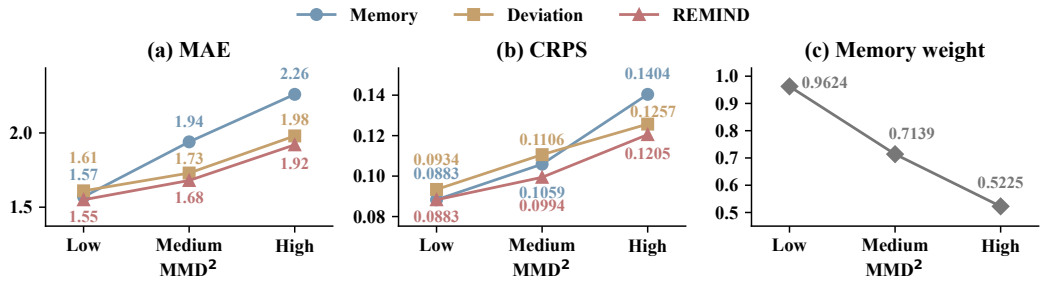

Figure 6: Branch performance and memory reliance across MMD^2 groups on ENERGY.

We analyze whether the gate responds to retrieval similarity and memory dispersion, and adjusts branch contributions accordingly under distribution shift.

Fig. 5 shows that the gate generally assigns greater memory weight at higher retrieval similarity and lower memory dispersion. This relationship is not strictly monotonic, especially for dispersion, consistent with flexible weighting that combines both signals under soft regularization.

Fig. 6 groups ENERGY test samples into tertiles by squared maximum mean discrepancy (MMD^2), computed for each sample against the training data using a Gaussian kernel. Higher MMD^2 indicates greater distribution shift. The memory branch performs better under milder shifts, whereas the deviation branch performs better under stronger shifts, accompanied by decreasing memory weights. Fusion achieves the lowest MAE in all groups and improves or matches both branches in CRPS.

Table 5 further examines samples with high retrieval similarity. As dispersion increases, the better-performing branch shifts from memory to deviation, and the gate accordingly reduces memory weight, demonstrating the role of memory dispersion in assessing retrieval reliability beyond similarity alone. Fusion consistently outperforms both branches, highlighting their complementary strengths and the effectiveness of adaptive fusion.

6 Conclusion

We propose REMIND, a retrieval-guided probabilistic framework for spatiotemporal forecasting. By leveraging retrieval as shift degree signals, REMIND adaptively balances memory reuse and deviation aware prediction, improving the generalization of probabilistic forecasting under distribution shift.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62372031).

References

[1] Shengnan Guo, Youfang Lin, Letian Gong, Chenyu Wang, Zeyu Zhou, Zekai Shen, Yiheng Huang, and Huaiyu Wan. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1585–1596. IEEE, 2023.

[2] 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, 2021.

[3] Zekai Shen, Haitao Yuan, Xiaowei Mao, Congkang Lv, Shengnan Guo, Youfang Lin, and Huaiyu Wan. Towards an efficient and effective en route travel time estimation framework. In *International Conference on Database Systems for Advanced Applications*, pages 604–619. Springer, 2025.

[4] Jun Rentschler, Melda Salhab, and Bramka Arga Jafino. Flood exposure and poverty in 188 countries. *Nature communications*, 13(1):3527, 2022.

[5] Haoyu Zhang, Wentao Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. Strap: Spatio-temporal pattern retrieval for out-of-distribution generalization. *arXiv preprint arXiv:2505.19547*, 2025.

[6] Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, Zhulin An, Qi Wang, and Yongjun Xu. Ginar+: A robust end-to-end framework for multivariate time series forecasting with missing values. *IEEE Transactions on Knowledge and Data Engineering*, 37(8):4635–4648, 2025.

[7] Fang Cheng, Hui Liu, and Xinwei Lv. Metagnsdformer: Meta-learning enhanced gated non-stationary informer with frequency-aware attention for point-interval remaining useful life prediction of lithium-ion batteries. *Advanced Engineering Informatics*, 69:103798, 2026.

[8] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

[9] Yang An, Zhibin Li, Wei Liu, Haoliang Sun, Meng Chen, Wenpeng Lu, and Yongshun Gong. Spatio-temporal graph normalizing flow for probabilistic traffic prediction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 45–55, 2024.

[10] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–12, 2023.

[11] Dongxia Wu, Liyao Gao, Matteo Chinazzi, Xinyue Xiong, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Quantifying uncertainty in deep spatiotemporal forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1841–1851, 2021.

[12] Xiyuan Jin, Jing Wang, Shengnan Guo, Tonglong Wei, Yiji Zhao, Youfang Lin, and Huaiyu Wan. Spatial–temporal uncertainty-aware graph networks for promoting accuracy and reliability of traffic forecasting. *Expert Systems with Applications*, 238:122143, 2024.

[13] Haochen Lv, Yan Lin, Shengnan Guo, Xiaowei Mao, Hong Nie, Letian Gong, Youfang Lin, and Huaiyu Wan. Ripcn: A road impedance principal component network for probabilistic traffic flow forecasting. *arXiv preprint arXiv:2512.21685*, 2025.

[14] Chengqing Yu, Fei Wang, Chuanguang Yang, Zezhi Shao, Tao Sun, Tangwen Qian, Wei Wei, Zhulin An, and Yongjun Xu. Merlin: Multi-view representation learning for robust multivariate time series forecasting with unfixed missing rates. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3633–3644, 2025.

[15] Yanru Sun, Zongxia Xie, Emadeldeen Eldele, Dongyue Chen, Qinghua Hu, and Min Wu. Learning pattern-specific experts for time series forecasting under patch-level distribution shift. *arXiv preprint arXiv:2410.09836*, 2024.

[16] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35:9881–9893, 2022.

[17] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4):1748–1764, 2021.

[18] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4396–4415, 2022.

[19] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[20] Zhiyuan Zhao, Haoxin Liu, and B Aditya Prakash. Tackling time-series forecasting generalization via mitigating concept drift. *arXiv preprint arXiv:2510.14814*, 2025.

[21] Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. *arXiv preprint arXiv:2505.04163*, 2025.

[22] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020.

[23] Yuefei Wu, Bin Shi, Bo Dong, Qinghua Zheng, and Hua Wei. The evidence contraction issue in deep evidential regression: Discussion and solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21726–21734, 2024.

[24] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[25] Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. Pristi: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1927–1939. IEEE, 2023.

[26] Yang An, Zhibin Li, Xiaoyu Li, Wei Liu, Xinghao Yang, Haoliang Sun, Meng Chen, Yu Zheng, and Yongshun Gong. Spatio-temporal multivariate probabilistic modeling for traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[27] Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2948–2959, 2024.

[28] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. *Advances in Neural Information Processing Systems*, 37:2766–2786, 2024.

[29] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[32] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[33] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[34] Nans Addor, Andrew J Newman, Naoki Mizukami, and Martyn P Clark. The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313, 2017.

[35] Xu Chen, Junshan Wang, and Kunqing Xie. Trafficstream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. *arXiv preprint arXiv:2106.06273*, 2021.

[36] Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022. *arXiv preprint arXiv:2208.04360*, 2022.

[37] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] Wei Chen and Yuxuan Liang. Expand and compress: Exploring tuning principles for continual spatio-temporal graph forecasting. *arXiv preprint arXiv:2410.12593*, 2024.

A Derivation of the Hierarchical Gaussian Mixture

In this appendix, we provide a detailed derivation of the hierarchical Gaussian mixture introduced in the Distribution Aware Fusion section of the main paper. We first show that the gate-weighted combination of the two branch-wise mixtures is itself a flat Gaussian mixture, then derive its predictive mean and variance, and finally write out the explicit form of the negative log-likelihood used as the training objective.

Notational conventions. The trust gate $\boldsymbol{g} \in [0,1]^{T\times N\times F}$ used in the gate-weighted fusion is applied element-wise, as are all branch-level parameters $(\boldsymbol{\mu}_{\mathrm{mem}}, \boldsymbol{\sigma}_{\mathrm{mem}})$, $(\boldsymbol{\mu}_{\mathrm{dev}}, \boldsymbol{\sigma}_{\mathrm{dev}})$, and the per-component residuals $(\Delta\boldsymbol{\mu}_i, \Delta\boldsymbol{\sigma}_i)$. Consequently, the predictive distribution $p(\boldsymbol{X}_P \mid \boldsymbol{X}_H)$ factorizes element-wise across spatial, temporal, and feature indices, and every quantity below is to be read as one such scalar entry. With this convention, we suppress the conditioning input $\boldsymbol{X}_H$ and the indices t, n, f throughout the derivation. The multivariate spatiotemporal case follows directly by stacking the per-element results. Two notational points are worth emphasizing to avoid clashes with the main paper. First, $\boldsymbol{\sigma}_i^{\mathrm{mem}} = \boldsymbol{\sigma}_{\mathrm{mem}} + \Delta\boldsymbol{\sigma}_i^{\mathrm{mem}}$ denotes a *component-level standard deviation*. Second, we use $\bar{\boldsymbol{\sigma}}^2_{\mathrm{mem}}$ to denote the *branch-level predictive variance* of p_{mem}. These two quantities are different objects and should not be conflated.

A.1 From a Hierarchical Mixture to a Flat Mixture

Recall from the main paper that the memory-calibrated branch and the deviation-aware branch are parameterized as Gaussian mixtures with I_{mem} and I_{dev} components, respectively:

$$p_{\mathrm{mem}}(\boldsymbol{X}_P \mid \boldsymbol{X}_H) = \sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{\pi}_i^{\mathrm{mem}} \mathcal{N}\big(\boldsymbol{X}_P \,\big|\, \boldsymbol{\mu}_i^{\mathrm{mem}}, (\boldsymbol{\sigma}_i^{\mathrm{mem}})^2\big)\,, \tag{18}$$

$$p_{\mathrm{dev}}(\boldsymbol{X}_P \mid \boldsymbol{X}_H) = \sum_{j=1}^{I_{\mathrm{dev}}} \boldsymbol{\pi}_j^{\mathrm{dev}} \mathcal{N}\big(\boldsymbol{X}_P \,\big|\, \boldsymbol{\mu}_j^{\mathrm{dev}}, (\boldsymbol{\sigma}_j^{\mathrm{dev}})^2\big)\,, \tag{19}$$

where, consistent with the main paper,

$$\boldsymbol{\mu}_i^{\mathrm{mem}} = \boldsymbol{\mu}_{\mathrm{mem}} + \Delta\boldsymbol{\mu}_i^{\mathrm{mem}}, \quad \boldsymbol{\sigma}_i^{\mathrm{mem}} = \boldsymbol{\sigma}_{\mathrm{mem}} + \Delta\boldsymbol{\sigma}_i^{\mathrm{mem}}, \tag{20}$$

$$\boldsymbol{\mu}_j^{\mathrm{dev}} = \boldsymbol{\mu}_{\mathrm{dev}} + \Delta\boldsymbol{\mu}_j^{\mathrm{dev}}, \quad \boldsymbol{\sigma}_j^{\mathrm{dev}} = \boldsymbol{\sigma}_{\mathrm{dev}} + \Delta\boldsymbol{\sigma}_j^{\mathrm{dev}}, \tag{21}$$

and the component weights satisfy $\sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{\pi}_i^{\mathrm{mem}} = 1$ and $\sum_{j=1}^{I_{\mathrm{dev}}} \boldsymbol{\pi}_j^{\mathrm{dev}} = 1$.

Substituting Eq. (18) and Eq. (19) into the gate-weighted fusion $p(\boldsymbol{X}_P \mid \boldsymbol{X}_H) = \boldsymbol{g}\, p_{\mathrm{mem}} + (1 - \boldsymbol{g})\, p_{\mathrm{dev}}$ defined in the main paper yields

$$\begin{aligned} p(\boldsymbol{X}_P \mid \boldsymbol{X}_H) &= \boldsymbol{g}\, p_{\mathrm{mem}}(\boldsymbol{X}_P \mid \boldsymbol{X}_H) + (1 - \boldsymbol{g})\, p_{\mathrm{dev}}(\boldsymbol{X}_P \mid \boldsymbol{X}_H) \\ &= \sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{g}\, \boldsymbol{\pi}_i^{\mathrm{mem}} \mathcal{N}\big(\boldsymbol{X}_P \,\big|\, \boldsymbol{\mu}_i^{\mathrm{mem}}, (\boldsymbol{\sigma}_i^{\mathrm{mem}})^2\big) \\ &\quad + \sum_{j=1}^{I_{\mathrm{dev}}} (1 - \boldsymbol{g})\, \boldsymbol{\pi}_j^{\mathrm{dev}} \mathcal{N}\big(\boldsymbol{X}_P \,\big|\, \boldsymbol{\mu}_j^{\mathrm{dev}}, (\boldsymbol{\sigma}_j^{\mathrm{dev}})^2\big)\,. \end{aligned} \tag{22}$$

Eq. (22) shows that the hierarchical mixture is equivalent to a flat Gaussian mixture with $I_{\mathrm{mem}} + I_{\mathrm{dev}}$ components, whose effective mixing weights are

$$\tilde{\boldsymbol{\pi}}_i = \boldsymbol{g}\, \boldsymbol{\pi}_i^{\mathrm{mem}} \;\; (i = 1, \ldots, I_{\mathrm{mem}}), \qquad \tilde{\boldsymbol{\pi}}_{I_{\mathrm{mem}}+j} = (1 - \boldsymbol{g})\, \boldsymbol{\pi}_j^{\mathrm{dev}} \;\; (j = 1, \ldots, I_{\mathrm{dev}}). \tag{23}$$

The effective weights remain a valid probability distribution, since

$$\sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{g}\, \boldsymbol{\pi}_i^{\mathrm{mem}} + \sum_{j=1}^{I_{\mathrm{dev}}} (1 - \boldsymbol{g})\, \boldsymbol{\pi}_j^{\mathrm{dev}} = \boldsymbol{g} + (1 - \boldsymbol{g}) = 1. \tag{24}$$

This flattening view is convenient for analysis and for evaluating the log-likelihood, while the hierarchical view of the gate-weighted fusion is more interpretable. The inner level captures intra-branch multimodality, and the outer level captures distribution-aware adaptation between the two branches.

A.2 Predictive Mean

We first compute the *branch-level* predictive mean of each branch from its own mixture in Eq. (18) and Eq. (19). To distinguish these from the component-level means $\boldsymbol{\mu}_i^{\mathrm{mem}}, \boldsymbol{\mu}_j^{\mathrm{dev}}$, we denote them with a bar:

$$\bar{\boldsymbol{\mu}}_{\mathrm{mem}} \triangleq \mathbb{E}_{p_{\mathrm{mem}}}[\boldsymbol{X}_P] = \sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{\pi}_i^{\mathrm{mem}}\, \boldsymbol{\mu}_i^{\mathrm{mem}}, \qquad \bar{\boldsymbol{\mu}}_{\mathrm{dev}} \triangleq \mathbb{E}_{p_{\mathrm{dev}}}[\boldsymbol{X}_P] = \sum_{j=1}^{I_{\mathrm{dev}}} \boldsymbol{\pi}_j^{\mathrm{dev}}\, \boldsymbol{\mu}_j^{\mathrm{dev}}. \tag{25}$$

Taking the expectation over the flat mixture in Eq. (22) and using linearity, the overall predictive mean becomes a gate-weighted combination of the two branch means:

$$\mathbb{E}[\boldsymbol{X}_P] = \boldsymbol{g}\, \bar{\boldsymbol{\mu}}_{\mathrm{mem}} + (1 - \boldsymbol{g})\, \bar{\boldsymbol{\mu}}_{\mathrm{dev}}. \tag{26}$$

A.3 Predictive Variance

For the variance, we again work at the branch level. We define the *branch-level predictive variance* as $\bar{\boldsymbol{\sigma}}^2_{\mathrm{mem}} \triangleq \mathrm{Var}_{p_{\mathrm{mem}}}[\boldsymbol{X}_P]$ and analogously for the deviation branch. This is an aggregate quantity over all I_{mem} components and should not be confused with any single $(\boldsymbol{\sigma}_i^{\mathrm{mem}})^2$. Using the standard identity $\mathrm{Var}[Z] = \mathbb{E}[Z^2] - (\mathbb{E}[Z])^2$ for a mixture, we obtain

$$\bar{\boldsymbol{\sigma}}^2_{\mathrm{mem}} = \sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{\pi}_i^{\mathrm{mem}} \big[(\boldsymbol{\sigma}_i^{\mathrm{mem}})^2 + (\boldsymbol{\mu}_i^{\mathrm{mem}})^2\big] - \bar{\boldsymbol{\mu}}^2_{\mathrm{mem}}, \tag{27}$$

$$\bar{\boldsymbol{\sigma}}^2_{\mathrm{dev}} = \sum_{j=1}^{I_{\mathrm{dev}}} \boldsymbol{\pi}_j^{\mathrm{dev}} \big[(\boldsymbol{\sigma}_j^{\mathrm{dev}})^2 + (\boldsymbol{\mu}_j^{\mathrm{dev}})^2\big] - \bar{\boldsymbol{\mu}}^2_{\mathrm{dev}}. \tag{28}$$

Applying the same identity to the flat mixture in Eq. (22) gives

$$\mathrm{Var}[\boldsymbol{X}_P] = \boldsymbol{g}\big(\bar{\boldsymbol{\sigma}}^2_{\mathrm{mem}} + \bar{\boldsymbol{\mu}}^2_{\mathrm{mem}}\big) + (1 - \boldsymbol{g})\big(\bar{\boldsymbol{\sigma}}^2_{\mathrm{dev}} + \bar{\boldsymbol{\mu}}^2_{\mathrm{dev}}\big) - \big(\boldsymbol{g}\, \bar{\boldsymbol{\mu}}_{\mathrm{mem}} + (1 - \boldsymbol{g})\, \bar{\boldsymbol{\mu}}_{\mathrm{dev}}\big)^2. \tag{29}$$

Expanding the square and collecting terms yields the following more interpretable decomposition:

$$\mathrm{Var}[\boldsymbol{X}_P] = \boldsymbol{g}\, \bar{\boldsymbol{\sigma}}^2_{\mathrm{mem}} + (1 - \boldsymbol{g})\, \bar{\boldsymbol{\sigma}}^2_{\mathrm{dev}} + \boldsymbol{g}(1 - \boldsymbol{g})\, \big(\bar{\boldsymbol{\mu}}_{\mathrm{mem}} - \bar{\boldsymbol{\mu}}_{\mathrm{dev}}\big)^2. \tag{30}$$

Eq. (30) decomposes the predictive uncertainty into two sources. The first term is a gate-weighted average of the within-branch mixture variances and reflects how uncertain each branch is on its own. The second term captures the discrepancy between the memory-calibrated and deviation-aware branch means. It increases when the two branches disagree and reaches its largest value when the gate is most ambivalent, namely when $\boldsymbol{g} \to \frac{1}{2}$. As a result, the model reports higher uncertainty in the transitional regime between in-distribution and out-of-distribution inputs even when each branch is individually confident.

A.4 Explicit Form of the Negative Log-Likelihood

We now derive the explicit form of the negative log-likelihood referenced in the Training Objective section of the main paper namely the unified mixture form introduced after the definition of $\mathcal{L}_{\mathrm{NLL}}$. Substituting Eq. (22) directly into $\mathcal{L}_{\mathrm{NLL}} = -\,\mathbb{E}[\log p(\boldsymbol{X}_P \mid \boldsymbol{X}_H)]$ gives

$$\begin{aligned} \mathcal{L}_{\mathrm{NLL}} = -\,\mathbb{E}\Bigg[\log\Bigg(& \sum_{i=1}^{I_{\mathrm{mem}}} \boldsymbol{g}\, \boldsymbol{\pi}_i^{\mathrm{mem}}\, \mathcal{N}\big(\boldsymbol{X}_P \,\big|\, \boldsymbol{\mu}_i^{\mathrm{mem}}, (\boldsymbol{\sigma}_i^{\mathrm{mem}})^2\big) \\ & + \sum_{j=1}^{I_{\mathrm{dev}}} (1 - \boldsymbol{g})\, \boldsymbol{\pi}_j^{\mathrm{dev}}\, \mathcal{N}\big(\boldsymbol{X}_P \,\big|\, \boldsymbol{\mu}_j^{\mathrm{dev}}, (\boldsymbol{\sigma}_j^{\mathrm{dev}})^2\big)\Bigg)\Bigg]. \end{aligned} \tag{31}$$

For numerical stability, we evaluate Eq. (31) with the log-sum-exp trick. Let $\tilde{\boldsymbol{\pi}}_k$ and $\big(\tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\sigma}}_k^2\big)$ denote the effective weight and parameters of the k-th component of the flat mixture, where $k = 1, \ldots, I_{\mathrm{mem}} + I_{\mathrm{dev}}$. These quantities are defined in Eq. (23). Then

$$\mathcal{L}_{\mathrm{NLL}} = -\,\mathbb{E}\Big[\mathrm{LSE}_k\Big(\log \tilde{\boldsymbol{\pi}}_k - \tfrac{1}{2}\log\big(2\pi\tilde{\boldsymbol{\sigma}}_k^2\big) - \tfrac{(\boldsymbol{X}_P - \tilde{\boldsymbol{\mu}}_k)^2}{2\tilde{\boldsymbol{\sigma}}_k^2}\Big)\Big], \tag{32}$$

where $\mathrm{LSE}_k(\cdot) \triangleq \log\sum_k \exp(\cdot)$. Combining this with the auxiliary similarity- and dispersion-aware regularizers introduced in the main paper gives the full training objective $\mathcal{L} = \mathcal{L}_{\mathrm{NLL}} + \lambda(\mathcal{L}_{\mathrm{sim}} + \mathcal{L}_{\mathrm{dis}})$. Setting $\boldsymbol{g} \in \{0, 1\}$ in Eq. (26) and Eq. (30) reduces the model to a single-branch mixture, and setting $I_{\mathrm{mem}} = I_{\mathrm{dev}} = 1$ reduces it to a single-Gaussian forecaster.

B Discussion of Spatiotemporal Retrieval

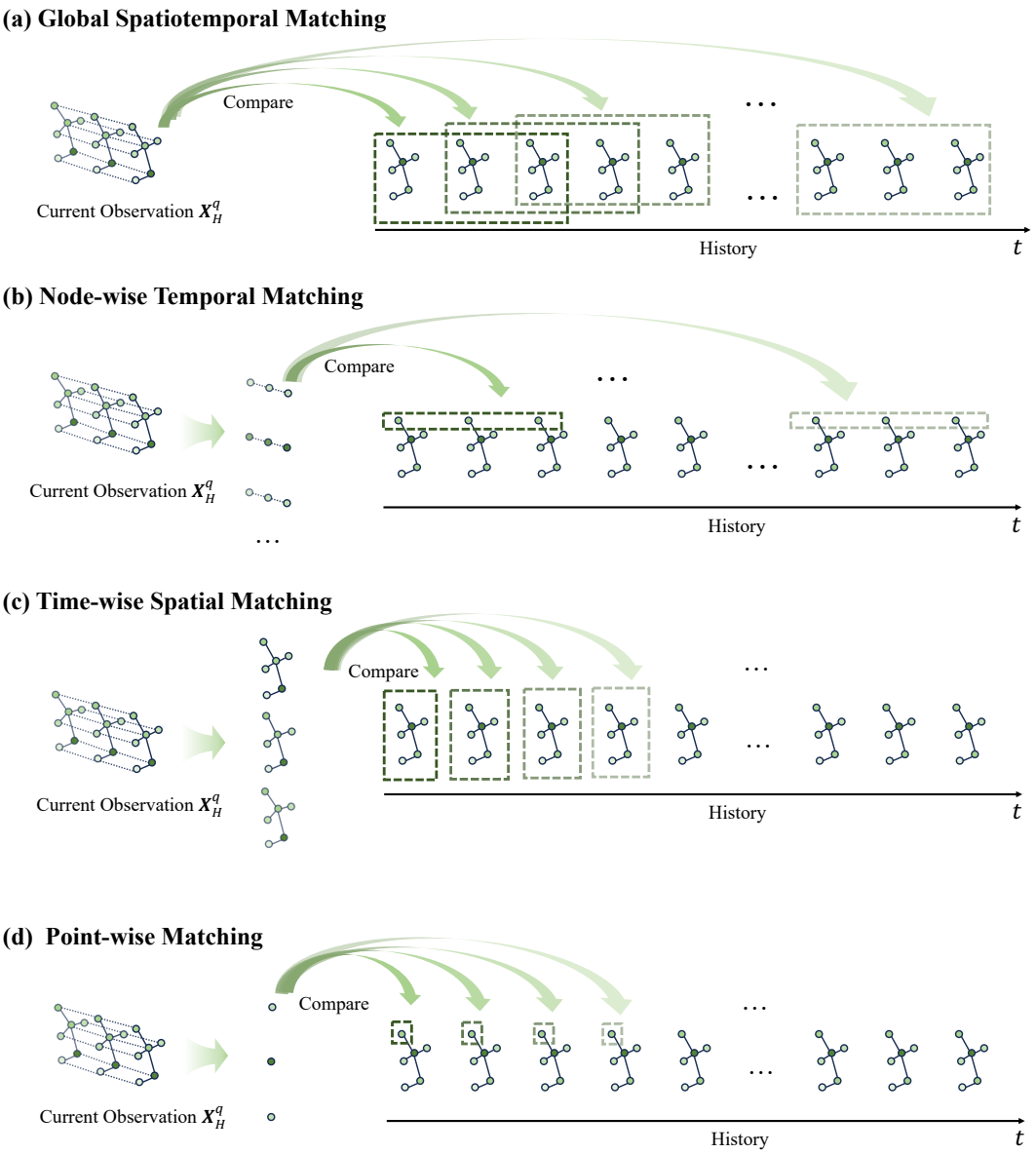

Figure 7: Four levels of spatiotemporal retrieval.

Let $\mathcal{D} = \{(\boldsymbol{X}_H^i, \boldsymbol{X}_P^i)\}_{i=1}^R$ denote a memory bank of R historical observation-target pairs. In an ideal spatiotemporal retrieval mechanism, given a query observation $\boldsymbol{X}_H^q$, the system aims to rapidly and efficiently retrieve the top K most similar samples $\mathcal{D}_{\mathrm{ret}} = \{\boldsymbol{X}_H^k, \boldsymbol{X}_P^k, \boldsymbol{s}_k\}_{k=1}^K$, where $\boldsymbol{s}_k$ denotes the similarity score between $\boldsymbol{X}_H^q$ and $\boldsymbol{X}_H^k$.

Retrieving spatiotemporal data is non-trivial due to the entanglement of spatial and temporal dimensions. Accordingly, spatiotemporal retrieval can be formulated at four levels: global spatiotemporal matching, node-wise temporal matching, time-wise spatial matching, and point-wise matching.

Table 6: Comparison of different spatiotemporal retrieval strategies.

Method	CAMELS			ENERGY			PEMS		
	MAE	MAPE	Time	MAE	MAPE	Time	MAE	MAPE	Time
Global	11.73	43.65	1.53	8.85	109.76	920.53	27.27	43.54	11407.35
Node-wise	10.87	36.40	3.90	7.96	104.70	3569.83	19.29	28.12	21179.26
Time-wise	14.00	58.26	7.15	8.76	108.58	467.50	27.41	44.42	4526.34
Point-wise	16.09	60.87	8.66	10.97	139.52	1239.79	41.72	53.20	15527.86
Ours	10.92	37.81	1.83	8.04	104.02	957.53	19.02	27.93	12377.27

Global spatiotemporal matching preserves overall structure with high efficiency but enforces strict temporal alignment across nodes. In contrast, node-wise temporal matching retrieves patterns independently for each node based on its temporal dynamics, enabling more flexible and fine-grained retrieval, but requiring independent matching for each node and thus increasing computational cost. Meanwhile, time-wise spatial matching disrupts temporal continuity, and point-wise matching ignores global spatiotemporal structure, limiting their ability to capture coherent spatiotemporal patterns.

Retrieving spatiotemporal data is challenging due to the strong coupling between spatial and temporal dependencies. Depending on how spatial and temporal structures are matched, spatiotemporal retrieval can be categorized into four levels: global spatiotemporal matching, node-wise temporal matching, time-wise spatial matching, and point-wise matching.

Global spatiotemporal matching treats the entire observation sequence as a unified spatiotemporal pattern and retrieves historical samples according to overall similarity. Given M historical samples and flattened feature dimension $D = T \times N \times F$, its retrieval complexity is approximately $\mathcal{O}(MD)$ for each query sample. This strategy preserves global structure and achieves high retrieval efficiency because matching is performed only once at the sample level. However, it enforces strict temporal alignment across all nodes, making it difficult to capture local temporal misalignment and heterogeneous regional dynamics under distribution shift.

Node-wise temporal matching retrieves historical patterns independently for each node based on temporal dynamics. For each query node, retrieval is performed against node-level candidates, resulting in a complexity of approximately $\mathcal{O}(NMD_t)$, where $D_t = T \times F$ denotes the temporal feature dimension for each node. Compared with global retrieval, this strategy relaxes strict node-level temporal alignment and enables more flexible reuse of local historical patterns, especially when different regions evolve asynchronously. This property is particularly important under distribution shift, where different spatial regions may shift at different times, such as localized traffic peaks caused by large events or regional disturbances. However, the independent retrieval process for each node substantially increases both computational and memory costs as the number of nodes grows.

Time-wise spatial matching performs retrieval independently at each time step based on spatial configurations. Although this strategy can model instantaneous spatial similarity, it breaks temporal continuity and may retrieve temporally inconsistent fragments from unrelated periods. For example, two observations with similar instantaneous values may correspond to completely different temporal states, such as rising and declining phases of system dynamics. As a result, coherent temporal evolution patterns are difficult to preserve.

Point-wise matching performs retrieval at the individual spatiotemporal point level. This strategy provides maximum flexibility and the finest retrieval granularity, but completely ignores higher-level spatiotemporal structure. Consequently, retrieved points may lack semantic consistency and fail to represent meaningful system-level dynamics.

Based on these observations, REMIND adopts a coarse-to-fine spatiotemporal retrieval strategy that combines global spatiotemporal retrieval with node-wise temporal retrieval. The first stage efficiently identifies globally relevant historical contexts, while the second stage refines local temporal matching and improves retrieval diversity under heterogeneous distribution shift. In the second stage, retrieval is further allowed across physically connected neighbors and statistically correlated nodes. This design is motivated by the spatial dependency and propagation characteristics of spatiotemporal systems: state transitions occurring at one node may gradually influence neighboring regions, allowing the model to capture relevant temporal dynamics from nearby nodes in advance. In addition, cross-node retrieval increases retrieval diversity and reduces over-reliance on exact node-level matches. However,

Table 7: Detailed statistics of datasets.

Dataset	Domain	#Nodes	#Steps	Resolution	Time Span	Feature
CAMELS	Hydrology	7	12699	1 day	1980-01-01 – 2014-10-07	Flow
PEMS	Traffic	581	106848	5 min	2011–2017 (07-10 to 08-31)	Flow
ENERGY	Energy	103	34560	10 min	240 days	Wind speed

retrieval is not performed across all nodes, since unrestricted matching would substantially increase computational cost and may introduce misleading patterns from spatially unrelated regions with significantly different dynamics.

As shown in Tab. 6, different retrieval strategies exhibit clear trade-offs between retrieval quality and computational efficiency. Global retrieval achieves the highest efficiency but suffers from limited flexibility due to strict temporal alignment across nodes. Node-wise retrieval substantially improves forecasting performance by enabling more flexible local temporal matching, but incurs significantly higher computational cost because retrieval must be performed independently for each node. Time-wise and point-wise retrieval both perform worse, confirming that breaking temporal continuity or ignoring higher-level spatiotemporal structure weakens the ability to capture coherent system dynamics. Overall, the proposed coarse-to-fine retrieval strategy achieves the best balance between retrieval quality and efficiency by combining globally consistent retrieval with locally adaptive refinement.

C Experimental Supplementary

C.1 Datasets

CAMELS [34], PEMS [35], and ENERGY [39, 36] are three real-world spatiotemporal datasets from hydrology, transportation, and energy domains, respectively. As shown Tab. 7, CAMELS contains daily streamflow observations from 7 basins over a long time span from 1980 to 2014, capturing long-term hydrological variability. In this work, we extract data from Basin 13 for evaluation. PEMS consists of traffic flow data collected at 5-minute intervals over 7 years from 581 sensors, reflecting strong spatiotemporal dynamics. ENERGY includes wind speed measurements at 10-minute resolution over 240 days from 103 locations, representing high-frequency variations in energy systems. As the monitored locations vary over time, we retain only those shared across the entire observation period for both datasets to ensure spatial consistency.

These datasets exhibit diverse temporal and spatial resolutions from minutes to days and long observation horizons, leading to pronounced distribution shifts across time and conditions, which pose significant challenges for reliable probabilistic forecasting.

C.2 Metrics

We evaluate REMIND by standard deterministic metrics, including MAE, RMSE, and MAPE, and uncertainty metrics, including Continuous Ranked Probability Score (CRPS) [37] and the Energy Score (ES) [9].

Continuous Ranked Probability Score (CRPS). CRPS is a proper scoring rule for evaluating probabilistic forecasts by measuring the discrepancy between the predicted cumulative distribution function and the observed value. It evaluates the entire predictive distribution rather than a single point estimate.

$$\text{CRPS}(F, x) = \int_{-\infty}^{\infty} \big[F(z) - \mathbb{I}\{x \le z\}\big]^2 dz, \tag{33}$$

where, $F(z)$ denotes the predicted cumulative distribution function, x is the ground-truth observation, $\mathbb{I}\{\cdot\}$ is the indicator function.

In practice, CRPS is approximated using a finite set of quantiles $\mathcal{Q} = \{0.05, 0.10, \ldots, 0.95\}$:

$$\text{CRPS}(y_{\text{true}}, y_{\text{pred}}) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{L_q(y_{\text{true}}, y_q^{\text{pred}})}{\sum |y_{\text{true}}|}, \tag{34}$$

$$L_q(y_{\text{true}}, y_q^{\text{pred}}) = 2 \sum \left| (y_q^{\text{pred}} - y_{\text{true}}) \cdot (\mathbb{I}[y_{\text{true}} \leq y_q^{\text{pred}}] - q) \right|, \tag{35}$$

where, y_{true} is the ground-truth value, y_{pred} denotes predictive samples, y_q^{pred} is the predicted q-th quantile, $L_q(\cdot)$ is the pinball loss, $|\mathcal{Q}|$ is the number of quantile levels.

Lower CRPS indicates better alignment between the predicted distribution and the ground truth.

Energy Score (ES). ES is a multivariate proper scoring rule derived from the energy distance, designed for evaluating probabilistic forecasts in high-dimensional and spatiotemporal settings. It simultaneously measures forecast accuracy and predictive dispersion.

$$\text{ES} = \mathbb{E}\big[||y_{\text{pred}} - y_{\text{true}}||_2\big] - \frac{1}{2}\mathbb{E}\big[||y_{\text{pred}}^{(1)} - y_{\text{pred}}^{(2)}||_2\big], \tag{36}$$

where, y_{pred} denotes predictive samples drawn from the forecast distribution, y_{true} is the ground-truth observation, $||\cdot||_2$ is the ℓ_2 norm over the spatial dimension, $y_{\text{pred}}^{(1)}$ and $y_{\text{pred}}^{(2)}$ are two independent samples from the predictive distribution.

The first term measures the expected distance between predictions and observations (accuracy), while the second term captures the diversity among predictive samples (uncertainty representation). Lower ES values indicate more accurate and well-calibrated probabilistic forecasts.

C.3 Implementation Details.

All experiments are implemented in PyTorch 2.4.0 and conducted on an NVIDIA GeForce RTX A4000 GPU. For each dataset, baseline hyperparameters are carefully tuned around their recommended or default settings to ensure optimal performance. All results are averaged over three runs with different random seeds. All datasets are split into training, validation, and test sets with a ratio of 6:2:2, and Z-score normalization is applied. The model predicts the next 12 time steps based on the previous 12 observations. We adopt the Adam optimizer [38] with a batch size of 12 and an initial learning rate of $1\text{e}{-4}$. The hidden dimension is set to 64, and the memory perturbation strength is set to 0.2. The number of retrieved samples K, the numbers of Gaussian components I_{mem} and I_{dev}, and the loss hyperparameter λ are determined via hyperparameter tuning.

C.4 Additional Comparison with Distribution Shift Methods

Table 8: Comparison with Distribution Shift Methods on PEMS

Metric	ShiftTS	STONE	TFPS	STRAP	**REMIND**
MAE	16.84±0.21	18.65±0.28	17.29±0.23	**16.60**±0.20	<u>16.63</u>±0.19
RMSE	27.82±0.35	30.53±0.42	30.31±0.40	**26.32**±0.34	<u>26.88</u>±0.32
MAPE	18.00±0.48	21.21±0.65	<u>17.69</u>±0.46	20.24±0.55	**17.76**±0.44

Table 9: Comparison with Distribution Shift Methods on ENERGY

Metric	ShiftTS	STONE	TFPS	STRAP	**REMIND**
MAE	<u>1.76</u>±0.03	2.35±0.04	1.96±0.03	2.77±0.05	**1.72**±0.03
RMSE	<u>2.43</u>±0.04	2.93±0.05	2.49±0.04	3.75±0.06	**2.39**±0.04
MAPE	23.82±0.55	<u>23.51</u>±0.58	22.16±0.50	26.28±0.70	**21.49**±0.48

Tabs. 8 and 9 provide additional comparisons with deterministic distribution shift methods on PEMS and ENERGY. REMIND remains highly competitive across both datasets, achieving the best or near-best deterministic performance while also preserving probabilistic forecasting capability. On ENERGY, REMIND consistently outperforms the compared shift-aware baselines, showing that its adaptive memory usage is effective under high-frequency dynamics where retrieved patterns can vary substantially over time.

On PEMS, REMIND performs on par with the strongest deterministic baseline and achieves the best result on one of the reported metrics. This is a strong result because the competing methods are

specifically optimized for point prediction under distribution shift, whereas REMIND additionally learns predictive uncertainty. The comparison suggests that modeling uncertainty and adapting memory reliance can provide a more stable forecasting strategy under non-stationary traffic patterns.

C.5 Additional Ablation Study

Table 10: Ablation study on core architecture.

Variant	MAE	RMSE	MAPE	CRPS	ES
w/o Mem	9.51	33.19	32.04	0.2992	32.75
w/o Dev	9.15	32.83	28.61	0.2309	26.79
w/o Gate	9.29	32.56	31.53	0.2439	29.01
REMIND	**8.73**	**32.26**	**25.30**	**0.2232**	**26.04**

Table 11: Ablation study on key mechanisms.

Variant	MAE	RMSE	MAPE	CRPS	ES
w/o MD	8.93	32.30	28.04	0.2546	30.06
w/o MP	8.86	32.29	27.21	0.2494	29.49
w/o GM	8.83	32.35	26.03	0.2405	28.20
REMIND	**8.73**	**32.26**	**25.30**	**0.2232**	**26.04**

Tabs. 10 and 11 report additional ablation results on CAMELS. The trends are consistent with the ENERGY ablations in the main paper: removing any core component weakens performance, confirming that the memory branch, deviation branch, and gate each play a necessary role. The memory branch is especially important for probabilistic quality on CAMELS, where long-term hydrological patterns can provide useful historical support but still require careful calibration. The deviation-aware branch and gate further improve robustness by preventing the model from blindly trusting retrieved futures when the current basin dynamics differ from historical patterns.

The detailed ablations also support the effectiveness of the proposed mechanisms. Memory dispersion(MD) provides an explicit measure of retrieval reliability and improves uncertainty estimation. Memory perturbation(MP) encourages the model to tolerate imperfect memory and reduces sensitivity to retrieval errors. The Gaussian mixture(GM) improves both accuracy and calibration by allowing the predictive distribution to represent multiple plausible hydrological responses. These results show that the gains of REMIND do not come from a single module, but from combining retrieval, shift-aware fusion, and flexible probabilistic modeling.

C.6 Additional Hyperparameter Study

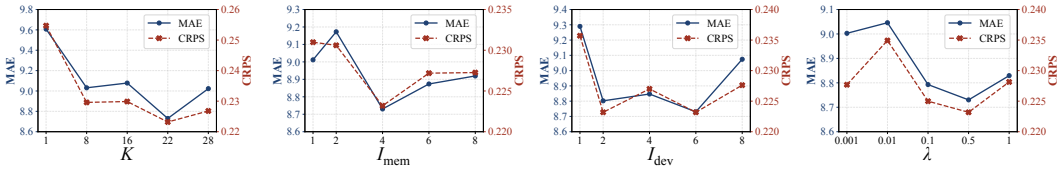

Figure 8: Hyperparameter sensitivity on CAMELS.

Fig. 8 shows the hyperparameter sensitivity on CAMELS. The overall trends are similar to ENERGY. A moderate top-K value performs best because it retrieves enough historical support without introducing excessive irrelevant memory. Too few retrieved samples make the prediction depend on narrow evidence, while too many samples dilute the relevance of the memory bank and increase inconsistency among retrieved futures.

For the Gaussian mixture configuration, the memory-calibrated branch prefers a compact number of components, while the deviation-aware branch benefits from a larger mixture. This again matches the model design: the memory branch already inherits possible modes from retrieved targets, whereas the deviation branch must construct a broader predictive distribution from deviation information. The sensitivity to λ also confirms the role of soft gate regularization. A moderate value gives the best balance between likelihood-driven learning and shift-signal guidance; overly weak regularization underuses retrieval statistics, while overly strong regularization restricts the adaptive behavior of the gate.

C.7 Robustness to Memory Corruption

To examine whether REMIND adapts when retrieved memory becomes less informative, we add Gaussian noise of increasing strength to retrieved samples at test time on ENERGY. Noise strength is

Table 12: Memory corruption on ENERGY.

	STRAP			REMIND			
Noise	MAE	RMSE	MAPE	MAE	RMSE	MAPE	Mean g
0.0	2.77	3.75	26.28	1.72	2.39	21.49	0.74
0.1	2.95	3.96	28.10	1.76	2.44	22.10	0.66
0.2	3.24	4.31	31.47	1.84	2.54	23.38	0.53
0.4	3.78	4.93	37.85	2.02	2.76	25.91	0.36
0.6	4.31	5.58	44.92	2.23	3.02	29.14	0.24

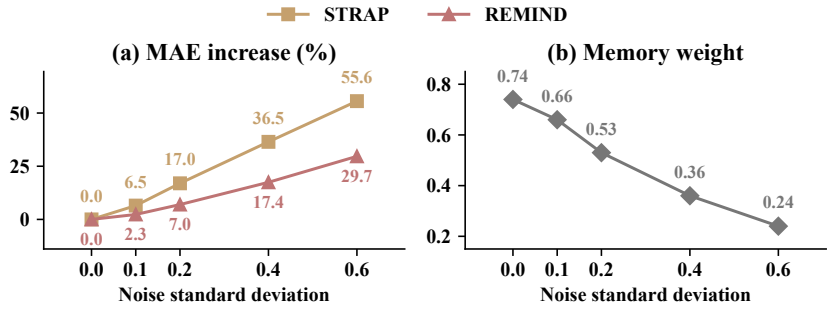

Figure 9: Robustness to memory corruption on ENERGY.

measured by its standard deviation in normalized data units. Table 12 reports forecasting errors for REMIND and STRAP, while Fig. 9 shows the relative MAE increase and REMIND's mean memory weight.

As noise increases, forecasting errors rise for both methods, but REMIND degrades more gradually. Its memory weight also decreases, indicating that the gate reduces direct reuse of corrupted historical targets. These observations are consistent with the intended adaptation: REMIND places less weight on retrieved targets as their quality declines, helping limit the effect of memory corruption on forecasting. This further highlights the importance of assessing retrieval reliability and adaptively adjusting memory utilization.

D Limitations and Broader Impacts

Although REMIND improves probabilistic forecasting under distribution shift, it still relies on the availability and quality of historical retrieval memory. Under extremely sparse, rapidly evolving, or entirely unseen scenarios, suitable historical patterns may be difficult to retrieve, which can limit the effectiveness of retrieval-guided adaptation.

For broader impacts, this work aims to improve the reliability and generalization of probabilistic spatiotemporal forecasting under non-stationary environments, which may benefit applications such as traffic management, energy systems, and hydrology. More reliable uncertainty estimation can support safer and more informed decision-making under evolving conditions. More broadly, this work highlights the potential of combining retrieval mechanisms with probabilistic modeling for robust decision support in dynamic spatiotemporal systems. However, under unseen scenarios, inaccurate or miscalibrated uncertainty estimates may still affect downstream decisions. Therefore, careful evaluation and responsible deployment remain important in real-world applications.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the scope and contributions of the paper. The proposed retrieval-guided probabilistic framework, shift-adaptive fusion mechanism, and hierarchical Gaussian mixture are formally described in Secs. 4.2-4.5 and further analyzed in the appendix. The claims regarding forecasting accuracy, probabilistic uncertainty modeling, and generalization under distribution shift are supported by the experimental results and ablation studies in Sec. 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of the proposed framework in App. D. In particular, we note that the effectiveness of retrieval-guided forecasting may depend on the availability of relevant historical patterns under complex distribution shifts.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The paper provides theoretical analysis of the hierarchical Gaussian mixture in App. A, including the derivation of the unified mixture formulation and predictive variance decomposition. All formulas are explicitly defined in the main paper and appendix. The derivations are presented with the required assumptions and intermediate steps to support the probabilistic interpretation of the proposed framework.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the proposed architecture, retrieval strategy, training objective, and experimental settings in Secs. 4-5 and the App. C. Dataset splits, evaluation metrics, optimization settings, hyperparameter selection, and implementation details are explicitly reported.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

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

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

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

   Answer: [Yes]

   Justification: The datasets used in this work are publicly available benchmark datasets. The paper provides an anonymous code repository in the abstract for reproducibility during the review process, and the code will be fully released upon acceptance. Detailed implementation settings, training configurations, and experimental protocols are also provided in Sec. 5 and App. C to support faithful reproduction of the main results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the proposed architecture, retrieval strategy, training objective, and experimental settings in Secs. 4-5 and the App. C. Dataset splits, evaluation metrics, optimization settings, hyperparameter selection, and implementation details are explicitly reported.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: As stated in Sec. 5.1, all experimental results are averaged over three runs with different random seeds. Standard deviations are reported in tables where space permits, reflecting variability caused by random initialization and training stochasticity. In addition, box plots with distribution quantiles are provided in Sec. 5 to further illustrate the statistical behavior of the proposed method.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports the experimental environment in Sec. 5.1, including the use of PyTorch 2.4.0 and an NVIDIA GeForce RTX A4000 GPU. App. B further reports the computational efficiency of different retrieval strategies and discusses their scalability under different retrieval granularities.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics. The work uses publicly available datasets in Sec. 5, preserves author anonymity during submission, and does not involve human subjects or sensitive personal data. The paper also discusses limitations and broader impacts of the proposed framework in App. D.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The paper discusses broader impacts in Sec. 1, which highlights the potential benefits of improving uncertainty modeling for spatiotemporal forecasting applications such as traffic management, energy systems, and hydrology. A more comprehensive discussion of both positive and potential negative impacts is further provided in App. D, including the potential influence of uncertainty estimation quality under unseen scenarios and the importance of careful evaluation in real-world applications.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The proposed method is a probabilistic spatiotemporal forecasting framework and does not involve high-risk generative models, scraped datasets, or sensitive personal data that would require special safeguards for release.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper uses publicly available benchmark datasets and properly cites the original sources of the datasets and baseline methods. The usage of these assets follows their publicly available terms and research usage policies. Relevant references and dataset descriptions are provided in Sec. 5 and the App.C.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: An anonymized code repository with usage documentation is provided in the abstract during the review process, and the full codebase and related documentation will be publicly released upon acceptance.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve research with human subjects or crowdsourcing experiments, and therefore no IRB approval is required.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core methodology and experiments in this paper do not involve LLMs as important, original, or non-standard components.

Guidelines:

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