EFFICIENT ZERO-SHOT SPATIO-TEMPORAL FORECASTING VIA MULTI-SCALE EXPERT DISTILLED MIXED HASHING ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Within-domain cross-dataset zero-shot spatio-temporal forecasting is crucial for applications such as smart cities and IoT, and while Large Language Models (LLMs) show promising capability in this setting, their high computational cost limits real-time deployment. Designing an efficient student model to distill knowledge from LLMs offers a promising solution, though it remains challenging. In this paper, we propose Multi-scale Expert Distilled Mixed Hashing Attention (MSED-MHA). We first employ Linear Attention to capture global spatio-temporal trends with high efficiency. In addition, we introduce Spectral Hashing Attention to model local dependencies through hashing-based Top-k retrieval over relevant neighbors. To avoid directly optimizing multiple discrete hash constraints, we construct an approximate spectral target to guide differentiable hash learning. In addition, a selective Multi-Scale Expert Distillation guides the student across temporal scales, applying teacher supervision only when beneficial, thereby mitigating error propagation. Experiments on traffic, finance, and weather benchmarks demonstrate that MSED-MHA achieves competitive forecasting performance with high computational efficiency compared to state-of-the-art methods.

1 INTRODUCTION

Spatio-temporal data, including traffic flow, energy consumption, and meteorological observations, exhibits complex temporal dynamics and spatial dependencies that are fundamental to modeling and forecasting in real-world systems (Chen et al., 2023; Zhou et al., 2021; Ma et al., 2022; Saad et al., 2024; Wu et al., 2023; Li et al., 2023). Traditional spatio-temporal forecasting methods focus on effectively integrating temporal and spatial dependencies to achieve accurate and efficient predictions (Sun et al., 2024b; Qin et al., 2023). However, in real-world applications such as smart cities, intelligent transportation, and IoT systems, models are often required to generalize to unseen environments without task-specific training, making zero-shot forecasting a practical necessity.

Recent advances in large language models (LLMs) and foundation models have demonstrated strong generalization ability in time-series forecasting (Liu et al., 2024a;b; Yin et al., 2025). However, most of these approaches are primarily designed for temporal dependency modeling and do not explicitly capture spatial relationships among variables (Liu et al., 2025f; Shi et al., 2025; Liu et al., 2025a; Ansari et al., 2024), which limits their effectiveness in spatio-temporal forecasting tasks. In contrast, zero-shot spatio-temporal forecasting requires capturing transferable spatial dependencies across different scenarios, which is inherently challenging due to the variability of spatial structures across domains. As a result, existing studies typically develop domain-specific solutions (e.g., in traffic or finance) by incorporating LLMs or domain priors to enhance generalization, often at the cost of increased model complexity. Therefore, achieving cross-scenario generalization in spatio-temporal forecasting typically relies on large-scale models with substantial computational overhead. This creates a critical challenge for real-world deployment, where applications such as edge devices, urban sensing systems, and real-time monitoring platforms operate under strict resource and latency constraints.

To address this issue, a natural approach is to design a lightweight yet effective student model and transfer the generalization capability of LLMs or foundation models via knowledge distillation.

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

Existing efficient attention mechanisms provide a useful foundation in this regard. Linear attention reduces the quadratic complexity of Transformers to linear time (Choromanski et al., 2021; Xiong et al., 2021; Zhang et al., 2025), enabling scalable modeling of long-range dependencies, but its inherent low-pass filtering property limits its ability to capture local high-frequency variations (Zhang et al., 2024a). In contrast, sparse attention mechanisms such as Top-k attention focus on the most relevant interactions, improving the modeling of local patterns, while potentially sacrificing global context. Hashing-based attention further accelerates this process through efficient nearest-neighbor search (Li et al., 2025; Gong et al., 2025), yet it often suffers from optimization challenges due to discrete hash constraints, which can lead to degraded code quality and unstable training. Meanwhile, existing distillation strategies typically enforce full imitation of the teacher (Tang et al., 2024; Liu et al., 2025c; Ni et al., 2025), which may propagate teacher errors and limit the student's flexibility. These limitations make it difficult to simultaneously achieve efficiency, robustness, and strong zero-shot generalization in spatio-temporal forecasting.

The above paradigm, which integrates a lightweight student model with efficient attention mechanisms and knowledge distillation, offers a feasible approach. However, achieving efficient and robust zero-shot spatio-temporal forecasting critically depends on two key aspects: an attention mechanism that can effectively balance global and local dependencies with low computational cost, and a distillation strategy that can transfer useful knowledge while avoiding the limitations of the teacher. These considerations naturally motivate the design of our method.

To address the aforementioned challenges, we propose a novel spatio-temporal forecasting method, named Multi-Scale Expert Distilled Mixer Hashing Attention (MSED-MHA). Our method incorporates two key components: the distillation strategy and the efficient student architecture. First, for the student architecture, we introduce a Mixed Hashing Attention (MHA) mechanism that combines the advantages of linear attention and hashing-based top-k attention. In MHA, linear attention serves as a global smoother that captures long-range trends, while the proposed Spectral Hashing Attention acts as a local refiner to focus on critical spatio-temporal events. Specifically, we improve hashing attention by constructing an approximate spectral target for the learnable hash function, thereby avoiding direct joint optimization of multiple discrete constraints, and subsequently employ a differentiable top-k mechanism to approximate discrete hash selection. Second, we design a selective Multi-Scale Expert Distillation (MSED) scheme. Spatio-temporal dynamics exhibit inherent multi-scale properties, ranging from short-term fluctuations to long-term trends. Our MSED scheme leverages this property by guiding the student model across different temporal scales and selectively applying teacher supervision when the student underperforms, thereby improving adaptability while reducing unnecessary distillation signals. This enables the student to better generalize and potentially surpass its teacher. Our main contributions are summarized as follows:

- We propose a Mixed Hashing Attention (MHA) that integrates Linear Attention for global trends with specifically designed Spectral Hashing Attention (SHA) for local refinement. SHA uses an approximate spectral target to guide differentiable hash learning without directly adding separate balance and orthogonality penalties to the training objective.
- We design a selective Multi-Scale Expert Distillation (MSED) mechanism, where experts within the distillation module act as scale-wise selectors that activate teacher guidance only when the student underperforms the teacher at the corresponding scale. This mechanism enables effective multi-scale learning while preventing the student from being limited by the teacher's performance.
- Comprehensive experiments on multiple public datasets demonstrate that our model achieves competitive within-domain cross-dataset zero-shot forecasting accuracy with low computational cost, yielding the best average ranks over the reported metrics.

2 Related Work

2.1 Spatio-Temporal Time Series Forecasting

Spatio-temporal forecasting aims to predict future trends in data with both temporal and spatial dependencies (Sun et al., 2024b), requiring effective integration of these two aspects. In contrast, multivariate time series forecasting mainly captures temporal and inter-variable dependencies without explicitly modeling structured spatial relationships. Recent studies have explored Transformer-based

architectures, such as PatchTST (Nie et al., 2023) and MTST (Zhang et al., 2024b), to capture long-range dependencies, along with multi-scale analysis methods that decompose signals into different resolutions (Murad et al., 2025; Wang et al., 2025b; Cai et al., 2024). At the same time, there is a growing emphasis on efficient lightweight models (Lin et al., 2026) and standardized benchmarks (Shao et al., 2025), reflecting the need for practical and fair evaluation.

Existing approaches to spatio-temporal forecasting mainly fall into two categories: Graph Neural Networks (GNNs) and Transformers. GNNs explicitly model spatial dependencies on graph structures (Kong et al., 2024; Ma et al., 2025), while Transformers capture long-range dependencies across both temporal and spatial dimensions (Chen & Zhao, 2024; Jin et al., 2025). Given that Large Language Models (LLMs) are built upon the Transformer architecture and demonstrate strong zero-shot generalization, adopting a Transformer-based student model is a natural choice for facilitating knowledge distillation in zero-shot forecasting. Therefore, this work focuses on designing an efficient Transformer-based architecture.

2.2 Large Language Models and Distillation for Spatio-temporal Data

Large Language Models (LLMs) have been widely applied in time series analysis, achieving strong performance in forecasting (Hu et al., 2025b), few-shot learning (Sun et al., 2024a), and anomaly detection (Zhou et al., 2023). Existing approaches can be broadly categorized into: (1) prompt-based methods that encode time series into designed prompts (Xue & Salim, 2024; Cao et al., 2024); (2) modality alignment techniques that bridge time series and text via reprogramming or cross-modal fine-tuning (Jin et al., 2024; Wang et al., 2025a; Liu et al., 2025d;e); and (3) in-context learning frameworks that enable adaptation from input samples (Lu et al., 2025).

These advances motivate extending LLMs to spatio-temporal forecasting. To capture complex topological dependencies, prior works propose specialized designs such as spatio-temporal embeddings, graph-to-token representations, and graph-aware modules (e.g., graph attention) (Liu et al., 2024a;b; 2025b). Some studies further integrate domain knowledge or combine LLMs with models such as GANs to enhance robustness (Jiang et al., 2025; Yin et al., 2025).

However, the large model size and high inference cost hinder practical deployment. Knowledge distillation addresses this by transferring knowledge from a teacher to a lightweight student. Existing methods explore spatio-temporal information bottlenecks (Tang et al., 2024), intermediate representation distillation (Liu et al., 2025c), and cross-architecture multi-scale alignment (Ni et al., 2025). Nevertheless, these approaches often enforce uniform imitation of the teacher, which may propagate errors and fail to adequately guide multi-scale spatio-temporal learning.

2.3 Efficient Attention Mechanisms

To alleviate the quadratic complexity of standard self-attention, linear attention reduces computation to linear cost via low-rank decomposition or kernel approximation. Representative methods include Performer (Choromanski et al., 2021) and Nyströmformer (Xiong et al., 2021), which approximate Softmax attention, followed by approaches such as Hedgehog (Zhang et al., 2024a) that aim to preserve its key properties, and more recent work like LoLCATs (Zhang et al., 2025) that extends linear attention to large language models.

Another line of research focuses on Top-k attention, which restricts each query to its most relevant keys to reduce memory complexity (Gupta et al., 2021; Wang et al., 2022; Zeng et al., 2025). A key challenge lies in efficiently identifying these keys without full dot-product computation. To address this, hashing-based methods have been proposed to accelerate similarity search through efficient, hardware-friendly operations (Li et al., 2025; Gong et al., 2025). We introduce this paradigm to spatio-temporal forecasting, as its locality is well-suited for capturing high-frequency burst patterns.

However, existing hashing methods suffer from notable limitations: they often lose global context and face optimization difficulties due to non-differentiable discrete codes. As a result, constraint relaxation is commonly adopted, which degrades hash quality and leads to unstable training.

3 Problem Definition

We formalize the spatio-temporal forecasting problem as inferring future values based on historical observations. Let $\mathcal{X} \in \mathbb{R}^{T \times N \times M}$ represent the input three-dimensional spatio-temporal tensor, where T, N, and M denote the length of the historical time window, the number of spatial nodes, and the number of features, respectively. The objective is to learn a mapping function f_θ to predict the signal for the next P time steps, denoted as $\mathcal{Y} \in \mathbb{R}^{P \times N \times M}$.

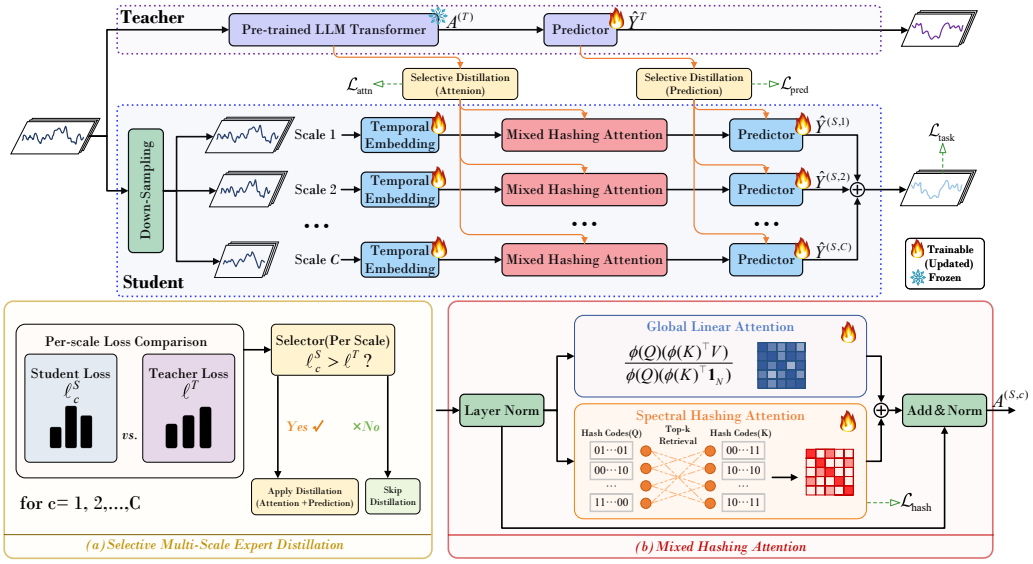

Figure 1: Overall framework of MSED-MHA, including (a) Selective Multi-Scale Expert Distillation for scale-wise conditional knowledge transfer and (b) Mixed Hashing Attention that combines global linear attention with spectral hashing-based Top-k attention to capture both global and local spatio-temporal patterns.

To capture common patterns across nodes and enhance the model's generalization capability, we adopt a Channel-Independent modeling strategy. Specifically, we decompose the input tensor $\mathcal{X}$ along the feature dimension into M independent spatio-temporal components $X^{(m)} \in \mathbb{R}^{T \times N}$, where $m \in \{1, \ldots, M\}$. The model f_θ, with shared parameters, is applied independently to each feature component:

$$\hat{Y}^{(m)} = f_\theta(X^{(m)}), \quad \text{where } \hat{Y}^{(m)} \in \mathbb{R}^{P \times N} \tag{1}$$

The final prediction result $\hat{\mathcal{Y}}$ is reconstructed by concatenating the predicted values of all feature components along the feature dimension:

$$\hat{\mathcal{Y}} = \text{Concat}(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(M)}) \in \mathbb{R}^{P \times N \times M} \tag{2}$$

4 Method

4.1 Overall Framework

Figure 1 illustrates the overall framework of MSED-MHA, where a lightweight student model performs multi-scale forecasting under the guidance of an LLM-based teacher. Our framework operates under a Teacher-Student paradigm, where a Teacher Model (e.g., a pre-trained large-scale spatio-temporal model) provides guidance to a lightweight Student Model. The student employs a multi-scale architecture: it first decomposes the inputs, then processes each scale with our Mixed Hashing Attention module, and finally fuses the outputs from parallel predictors. During training, the student is optimized end-to-end, learning from the ground truth while selectively absorbing

knowledge from the teacher via our distillation scheme. Crucially, the teacher is discarded entirely at inference time, allowing for efficient predictions to be generated by the student model alone.

4.2 Mixed Hashing Attention

To efficiently capture both long-range trends and local critical interactions in spatio-temporal data, we design a Mixed Hashing Attention. The output of this module is the sum of two attention components:

$$\text{MHA}(Q, K, V) = \text{GLA}(Q, K, V) + \text{SHA}(Q, K, V) \tag{3}$$

where $Q, K, V \in \mathbb{R}^{N \times d_{\text{model}}}$ represent the query, key, and value matrices, respectively. N is the number of spatial nodes, and d_{model} is the hidden feature dimension of the model. MHA first employs Global Linear Attention (GLA) to efficiently model long-range dependencies with linear computational complexity. In addition, Spectral Hashing Attention (SHA) is introduced to further capture local spatio-temporal interactions. Details of GLA are provided in Appendix A, while SHA is introduced in the following subsection.

4.3 Spectral Hashing Attention for Local Spatio-temporal Modeling

While linear attention effectively captures global, slowly varying trends in spatio-temporal data due to its low-pass filtering nature, it struggles to model local, high-frequency burst signals that are critical for precise forecasting. Real-world spatio-temporal associations are typically sparse, which motivates Top-k attention to focus on the most relevant neighbors while reducing computational redundancy. To further improve the efficiency of Top-k search, we propose Spectral Hashing Attention (SHA), which approximates Top-k selection through hashing-based retrieval with a spectral learning strategy. Specifically, SHA identifies and strengthens local associations among the most influential nodes, thereby effectively modeling high-frequency, bursty spatio-temporal dynamics.

4.3.1 Top-k Attention with Hashing

Our SHA improves Top-k attention by hashing technology. For each query q, it identifies the k most relevant keys by comparing their hash codes. The attention for q is computed only over the selected subset of keys and values, which constitute the sparsified key and value matrices, denoted as K' and V'. The Top-k attention for q can then be computed as:

$$\text{SHA}(q, K, V) = \text{Softmax}\left(\frac{q(K')^\top}{\sqrt{d_k}}\right) V' \tag{4}$$

Efficient Top-k attention relies critically on the construction of an appropriate hash function. For a query or key vector z, we define the hash logit $L_h(z) = zW_h + b_h$, where $W_h \in \mathbb{R}^{d_{\text{model}} \times b}$, $b_h \in \mathbb{R}^{1 \times b}$, and b is the hash length. The continuous training code and hard inference code are

$$h_c(z) = \tanh\big(\big(\beta L_h(z)\big)\big), \qquad h_d(z) = \text{sign}\big(L_h(z)\big). \tag{5}$$

Here, β controls the steepness of the continuous approximation, and we set $\beta = 10$ in all experiments.

4.3.2 Spectral Hashing Learning

Let $Z \in \mathbb{R}^{n \times d_{\text{model}}}$ denote the key matrix for one attention instance, and let $A = ZZ^\top$ and $R = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. We first form the centered matrix $Z_c = RZ$ and the spectral matrix

$$M = Z_c^\top A Z_c. \tag{6}$$

The implementation averages M across the mini-batch and takes the b eigenvectors associated with its largest eigenvalues as the columns of W. It then constructs the approximate spectral target $B = ZW$ and learns the hash projection by minimizing

$$\mathcal{L}_{\text{hash}}(W_h, b_h) = \|H_c - B\|_F^2, \qquad H_c = \tanh\big(\big(\beta L\big)\big), \quad L = ZW_h + \mathbf{1}b_h. \tag{7}$$

The eigenvector matrix W is treated as a stop-gradient quantity. The target $B = ZW$ still depends on the current key representation Z, matching the implemented computation.

Proposition 1. *For the symmetric matrix M, the matrix W containing its b leading orthonormal eigenvectors solves*

$$\max_{W} \quad \mathrm{Tr}(W^\top M W) \tag{8}$$

$$\textit{s.t.} \quad W^\top W = I. \tag{9}$$

The resulting $B = ZW$ is an approximate spectral target. In general, $W^\top W = I$ does not imply that B has zero orthogonality or balance residuals.

The derivation of Proposition 1 is provided in Appendix B.1. To state the residual transfer result at a scale compatible with binary codes, define $\bar{B} = B/\sqrt{n}$, $\bar{H}_c = H_c/\sqrt{n}$, and $\bar{\mathbf{1}} = \mathbf{1}/\sqrt{n}$.

Proposition 2. *Suppose the normalized spectral target satisfies $\|\bar{B}^\top \bar{B} - I\|_F \le \varepsilon_{\mathrm{orth}}$ and $\|\bar{B}^\top \bar{\mathbf{1}}\|_2 \le \varepsilon_{\mathrm{bal}}$, and let $\|\bar{H}_c - \bar{B}\|_F \le \delta_c$. Then $\|\bar{H}_c^\top \bar{H}_c - I\|_F \le \varepsilon_{\mathrm{orth}} + 2\delta_c\sqrt{1+\varepsilon_{\mathrm{orth}}} + \delta_c^2$ and $\|\bar{H}_c^\top \bar{\mathbf{1}}\|_2 \le \varepsilon_{\mathrm{bal}} + \delta_c$.*

Proposition 2 is conditional on the residuals of the approximate spectral target; it does not claim that the implementation makes them vanish. For the hard inference code $H_d = \mathrm{sign}(L)$, the corresponding bounds additionally depend on the normalized quantization gap between H_c and H_d. The proof and the margin-conditioned extension are provided in Appendix B.2; we do not claim unconditional stability under arbitrary domain shifts.

4.3.3 Differentiable Soft Top-k Attention

To enable end-to-end optimization, we relax the discrete Top-k selection into a differentiable approximation by constructing a soft weighting mechanism that preserves sparsity while remaining continuous. Given hash-based similarity scores, the proposed method generates a learnable weight matrix $\hat{W}$ to softly emphasize the most relevant key-value pairs, thereby approximating Top-k attention in a differentiable manner compatible with gradient-based training.

The final Soft Top-k Attention is formulated as:

$$\mathrm{SHA}(Q, K, V) = \mathrm{Softmax}\left(\frac{Q(\hat{W}K)^\top}{\sqrt{d_k}}\right)(\hat{W}V) \tag{10}$$

The detailed construction of $\hat{W}$ is provided in Appendix C.

4.4 Selective Multi-Scale Expert Distillation

To effectively capture heterogeneous temporal patterns in spatio-temporal data, we integrate multi-scale modeling with a selective knowledge distillation framework. The student model is designed as a multi-scale architecture, where the input sequence is decomposed into C temporal scales using non-overlapping average pooling with different strides, producing representations $\{X^{(c)}\}_{c=0}^{C-1}$. Each scale is processed through a temporal embedding module followed by a spatio-temporal modeling block based on MHA, enabling joint learning of temporal dynamics and spatial dependencies at different resolutions.

Formally, for each scale, the model produces a prediction $\hat{Y}^{(S,c)} \in \mathbb{R}^{P\times N}$, and the final forecast is obtained through weighted fusion:

$$\hat{Y} = \sum_{c=0}^{C-1} \omega_c \hat{Y}^{(S,c)}, \quad \omega_c = 1/C \tag{11}$$

Given this multi-scale design, the primary learning objective of the student is the Multi-Scale Task Loss, defined as the Smooth L1 distance between the fused prediction and ground truth: $\mathcal{L}_{\text{task}} = \ell_{\text{SL1}}(\hat{Y}, \mathcal{Y})$.

For each channel-independent input $X^{(m)} \in \mathbb{R}^{T\times N}$, the normalized T-step history of each node is treated as one numerical token. A shared linear layer maps each node history from $\mathbb{R}^T$ to the

GPT-2 hidden dimension $d_T = 768$, producing $E^T \in \mathbb{R}^{N \times d_T}$. These embeddings are passed through a frozen six-layer GPT-2 via `inputs_embeds`, without textual tokenization or prompting. A trainable forecasting head maps the final teacher representations $Z^T \in \mathbb{R}^{N \times d_T}$ to a P-step prediction $\hat{Y}^T \in \mathbb{R}^{P \times N}$. The teacher is used only during training and is discarded at inference.

We introduce a selective distillation mechanism instead of uniformly mimicking the teacher. Supervision is activated only when the student's performance at a given scale is inferior to that of the teacher. Specifically, let $\ell_c^S = \ell_{\text{SL1}}(\hat{Y}^{(S,c)}, \mathcal{Y})$ and $\ell^T = \ell_{\text{SL1}}(\hat{Y}^T, \mathcal{Y})$. The Selective Prediction Guidance Loss is defined as:

$$\mathcal{L}_{\text{pred}} = \sum_{c=0}^{C-1} \mathbb{I}(\ell_c^S > \ell^T) \cdot \ell_{\text{SL1}}(\hat{Y}^{(S,c)}, \hat{Y}^T) \tag{12}$$

Under the same scale-wise condition, we align the student's post-attention representation $Z^{(S,c)} \in \mathbb{R}^{N \times d_T}$ with the teacher's final hidden representation Z^T. Temporal downsampling changes the history resolution but not the number of node tokens, so no node-to-token projection is required. Thus, $\mathcal{L}_{\text{attn}}$ is a post-attention representation-alignment loss rather than a loss between raw attention-probability matrices:

$$\mathcal{L}_{\text{attn}} = \sum_{c=0}^{C-1} \mathbb{I}(\ell_c^S > \ell^T) \cdot \ell_{\text{SL1}}(Z^{(S,c)}, Z^T) \tag{13}$$

The final training objective is a weighted combination of all components, including the hash-based approximation loss $\mathcal{L}_{\text{hash}}$:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{attn}}\mathcal{L}_{\text{attn}} + \lambda_{\text{hash}}\mathcal{L}_{\text{hash}} \tag{14}$$

The complete training and inference procedures are provided in Appendix D.

Table 1: Summary of results for all methods on six datasets. The best results are in bold and the second best are underlined.

Model		MSED-MHA		TimeFilter		RePST		TimeCMA		FSCA		ST-LLM+		WPMixer		iTransformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS08	12	**0.1966**	**0.1988**	0.1992	0.2084	0.5287	0.4952	0.2407	0.2508	0.2053	0.2164	0.2059	0.2047	0.5939	0.3681	0.2366	0.2358
	24	**0.3069**	**0.2562**	0.3349	0.2802	0.6133	0.5252	0.3970	0.3357	0.3442	0.2824	0.3475	0.2737	0.9536	0.4499	0.3530	0.2961
	36	**0.4230**	**0.3123**	0.4804	0.3488	0.6835	0.5491	0.5597	0.4149	0.4863	0.3458	0.4917	0.3372	1.2785	0.5207	0.4761	0.3551
	48	**0.5243**	**0.3652**	0.6402	0.4148	0.7864	0.5892	0.7235	0.4864	0.6267	0.4054	0.6373	0.3958	1.5622	0.5829	0.5904	0.4108
SZ_TAXI	12	**0.2474**	**0.3319**	0.2762	0.3474	0.4428	0.4872	0.2576	0.3413	0.2542	0.3325	0.2686	0.3436	0.6409	0.5285	0.2837	0.3612
	24	**0.2582**	**0.3486**	0.3391	0.3899	0.4778	0.5028	0.3049	0.3733	0.3282	0.3761	0.3660	0.4030	0.9040	0.6272	0.3263	0.3886
	36	**0.2787**	**0.3599**	0.4048	0.4307	0.4871	0.5058	0.3568	0.4034	0.4120	0.4222	0.5178	0.4780	1.1104	0.6980	0.3688	0.4146
	48	**0.3092**	**0.3779**	0.4940	0.4801	0.4903	0.5057	0.4232	0.4368	0.5127	0.4754	0.7462	0.5721	1.3174	0.7635	0.3908	0.4279
METR-LA	12	0.3984	0.2982	0.3669	0.2954	0.4978	0.4212	0.3857	0.3144	**0.3401**	**0.2841**	0.3893	0.3016	0.9859	0.4973	0.4192	0.3379
	24	0.6217	0.3804	0.6179	0.4052	0.6722	0.4779	0.5709	0.3987	**0.5413**	**0.3759**	0.6463	0.4002	1.6264	0.6390	0.6093	0.4184
	36	0.8436	**0.4586**	0.8832	0.5057	0.7943	0.5181	**0.7157**	0.4644	0.7370	0.4616	0.9240	0.4875	2.2143	0.7531	0.8057	0.4958
	48	1.0927	0.5385	1.1611	0.5988	0.8844	0.5562	**0.8438**	**0.5199**	0.9174	0.5378	1.2414	0.5717	2.7044	0.8439	0.9436	0.5509
CSI300_2	12	1.0449	0.6589	1.2423	0.7397	**1.0268**	**0.6521**	1.0575	0.6655	1.0455	0.6592	1.4301	0.8293	1.2663	0.7450	1.3581	0.7808
	24	1.0346	0.6572	1.2290	0.7369	**1.0174**	**0.6516**	1.0504	0.6651	1.0349	0.6574	1.5997	0.8964	1.2593	0.7450	1.3333	0.7748
	36	1.0294	0.6565	1.2236	0.7365	**1.0130**	**0.6511**	1.0463	0.6649	1.0302	0.6570	1.5858	0.8875	1.2553	0.7456	1.3343	0.7765
	48	1.0200	0.6563	1.2171	0.7379	**1.0042**	**0.6506**	1.0368	0.6642	1.0210	0.6569	1.5419	0.8684	1.2431	0.7446	1.3221	0.7756
HSI_2	12	0.9686	0.5921	1.1457	0.6739	**0.9502**	**0.5856**	1.0027	0.6130	0.9695	0.5927	1.7156	0.8102	1.0786	0.6449	1.2534	0.7141
	24	0.9801	0.5926	1.1535	0.6725	**0.9648**	**0.5867**	1.0229	0.6167	0.9808	0.5930	2.0727	0.8969	1.0927	0.6460	1.2548	0.7099
	36	0.9960	0.5951	1.1688	0.6747	**0.9804**	**0.5891**	1.0425	0.6209	0.9968	0.5957	2.1102	0.9061	1.1109	0.6500	1.2828	0.7161
	48	1.0180	0.5979	1.1968	0.6797	**1.0017**	**0.5916**	1.0641	0.6233	1.0190	0.5986	2.0613	0.8941	1.1300	0.6516	1.3020	0.7179
weather2k	12	**0.5681**	**0.5460**	0.6751	0.5915	0.6662	0.5949	0.7631	0.6200	0.6272	0.5621	0.6487	0.5575	0.6831	0.5756	0.6352	0.5726
	24	**0.6087**	**0.5633**	0.7217	0.6090	0.6982	0.6090	0.8481	0.6346	0.7126	0.5996	0.8389	0.5921	0.7647	0.6090	0.7088	0.6073
	36	**0.6345**	**0.5737**	0.7548	0.6248	0.7198	0.6196	0.9310	0.6444	0.7565	0.6197	1.0068	0.6119	0.7945	0.6251	0.7482	0.6243
	48	**0.6510**	**0.5812**	0.7579	0.6274	0.7267	0.6234	0.9801	0.6516	0.7565	0.6300	1.1063	0.6230	0.7931	0.6281	0.7708	0.6351
avg rank		**2.0000**	**1.5417**	4.6667	5.0833	3.7500	4.6667	4.2917	4.5833	3.1250	3.0417	6.5000	5.0417	6.8750	6.5833	4.7917	5.4583

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

To evaluate the performance of MSED-MHA[1], we conducted experiments on fifteen spatio-temporal benchmark datasets covering three distinct domains: Traffic (PEMS03, PEMS04, PEMS07, PEMS08, NYC_TAXI, SZ_TAXI, METR-LA), Finance (Nikkei225, SP100, NASDAQ100, DJIA, CSI300, HSI), and Weather (weather, WTH, weather2k). Detailed information on these datasets is provided in Appendix E.1, and the full experimental settings are detailed in Appendix E.2.

[1]Our code is available at `https://anonymous.4open.science/r/MSED-MHA-anonym`

5.2 COMPARATIVE METHODS

Our experiments comprehensively compare our model's performance against several representative state-of-the-art (SOTA) models on zero-shot forecasting tasks. These baseline models cover multiple categories, including multivariate time-series models like WPMixer (Murad et al., 2025) and iTransformer (Liu et al., 2024c); spatio-temporal models such as TimeFilter (Hu et al., 2025a); and a series of LLM-based methods, including TimeCMA (Liu et al., 2025d), FSCA (Hu et al., 2025b), ST-LLM+ (Liu et al., 2025b) and RePST (Wang et al., 2025a). We use the official code released in the original papers and configure all baseline models following the experimental settings described therein.

5.3 MAIN RESULTS

Table 1 compares MSED-MHA with representative baselines under the within-domain cross-dataset zero-shot protocol across traffic, finance, and weather datasets. MSED-MHA does not lead every dataset, metric, and horizon, but it attains the best overall average ranks of 1.5417 for MAE and 2.0000 for MSE while retaining a substantially smaller deployed model than the LLM-based baselines. The results therefore support competitive accuracy across the evaluated domains, with particularly favorable performance in several long-horizon settings, rather than universal superiority in every individual comparison.

5.4 ABLATION STUDY

Table 2: Comparative performance of MSED-MHA and variants. The best results are in bold and the second best are underlined.

Methods		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
MSED-MHA	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	**0.2474**	**0.2582**	0.2787	0.3092	**0.3984**	0.6217	0.8436	1.0927
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	**0.3486**	**0.3599**	0.3779	**0.2982**	**0.3804**	**0.4586**	**0.5385**
w/o-Selective	MSE	0.3259	0.4113	0.5110	0.6300	0.2557	0.2644	**0.2749**	**0.3035**	0.4392	**0.5933**	**0.7834**	**1.0046**
	MAE	0.3131	0.3576	0.4024	0.4502	0.3548	0.3579	0.3615	**0.3770**	0.3474	0.4087	0.4787	0.5517
w/o-SHA	MSE	0.2111	0.3288	0.4542	0.6045	0.2894	0.3419	0.3836	0.4522	0.4062	0.6656	0.9507	1.2511
	MAE	0.2106	0.2665	0.3233	0.3830	0.3574	0.3899	0.4147	0.4533	0.3109	0.4035	0.4953	0.5888
Rep-SHA(Trad.)	MSE	0.2497	0.3923	0.5556	0.7275	0.2727	0.3542	0.4427	0.5524	0.4573	0.8326	1.3025	1.8043
	MAE	0.2331	0.2969	0.3580	0.4150	0.3516	0.4002	0.4483	0.5021	0.3374	0.4614	0.5852	0.6973

To systematically quantify the specific contributions of each core design within the MSED-MHA model, we conducted a series of detailed ablation experiments. We evaluated the performance changes by removing or replacing modules in MSED-MHA on three key traffic datasets. We designed the following model variants: 1) w/o-Selective, which removes the selective judgment mechanism and reverts to a mandatory, full knowledge transfer; 2) w/o-SHA, which completely removes the Spectral Hashing Attention module, causing the model to degenerate into a pure Global Linear Attention mechanism; and 3) Rep-SHA(Trad.), which replaces our spectral hashing with a traditional optimization method that uses penalty terms for constraints.

The ablation results in Table 2 support the roles of the core components. The degradation of w/o-Selective shows that mandatory imitation can transfer unfavorable teacher signals, while w/o-SHA confirms that global linear attention alone is insufficient in the evaluated settings. Rep-SHA(Trad.) further shows the difficulty of jointly optimizing multiple hash constraints through penalty terms. Additional controls in Appendix F isolate the two central effects more directly: selective distillation outperforms w/o-Distill in 23 of 24 traffic metric-horizon comparisons and reduces the average MSE and MAE by 19.9% and 13.6%, respectively (Table 5); spectral SHA outperforms an LSH replacement in 14 of 24 comparisons and reduces average MSE and MAE by 4.0% and 11.5%, respectively (Table 6).

For every target dataset, the integer $k = \lfloor N/30 \rfloor$ is recomputed from the target node count; neither an absolute source-domain k nor source-domain neighbor identities are reused. Appendix G.1 reports the corresponding sensitivity analysis.

Table 3: Experimental time consumption (seconds) and parameter analysis. The best results are in bold and the second best are underlined.

Methods	MSED-MHA		TimeFilter		RePST		TimeCMA		FSCA		ST-LLM+	
	Inference	Parameter	Inference	Parameter	Inference	Parameter	Inference	Parameter	Inference	Parameter	Inference	Parameter
PEMS08	81.39	13.51M	1376.59	**0.13M**	358.42	82.39M	491.21	136.94M	663.27	60.29M	**71.05**	82.41M
SZ_TAXI	**8.92**	13.51M	385.86	**0.13M**	223.11	82.39M	73.56	136.94M	103.83	60.29M	11.90	82.41M
METR-LA	**85.72**	13.51M	2443.73	**0.13M**	476.61	82.39M	917.62	136.94M	1235.06	60.29M	137.56	82.41M
weather2k	**486.22**	13.51M	5002.57	**0.12M**	1299.41	82.39M	18412.54	136.94M	4370.17	60.29M	698.02	82.35M

5.5 Efficiency Analysis

To evaluate the computational feasibility of MSED-MHA for practical deployment, especially in large-scale scenarios with a vast number of nodes (such as meteorological monitoring networks or financial markets), we analyzed its parameter count and inference latency. As shown in Table 3, we compared MSED-MHA with a series of representative models on several datasets, including weather2k, which contains 1,866 nodes.

The experimental results show that MSED-MHA provides a favorable accuracy-efficiency trade-off. For instance, on weather2k, its parameter count (13.51M) is about 9.8% of TimeCMA's (136.94M), while its inference time (486.22s) is about 37.9 times lower than TimeCMA's (18412.54s). TimeFilter uses fewer parameters, but its measured inference time is substantially higher. More detailed efficiency results are provided in Appendix H.

To assess run-to-run variability, Appendix I reports four-seed mean and standard deviation results for MSED-MHA and all traffic baselines. These results support the stability of the overall comparison while indicating that isolated small differences should not be overstated.

6 Conclusion

In this paper, we propose MSED-MHA for efficient within-domain cross-dataset zero-shot spatio-temporal forecasting. Our framework combines selective multi-scale distillation with Mixed Hashing Attention, whose spectral hashing component supports efficient local retrieval alongside global linear attention. Across the evaluated traffic, finance, and weather datasets, MSED-MHA achieves competitive accuracy and the best overall average ranks while using fewer parameters and lower inference time than the evaluated LLM-based approaches. The present study does not establish universal transfer across domains with different variable semantics and spatial structures. Extending the framework to such cross-domain settings while maintaining computational efficiency remains an important direction for future work.

Reproducibility Statement

We provide an anonymized code repository containing the implementation used in our experiments. Detailed experimental settings, hyperparameters, and dataset preprocessing procedures are provided in the main text and appendices.

AI Use Statement

We used generative AI tools for translating and polishing the manuscript. All AI-assisted text was carefully reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

References

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda-Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon

Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Trans. Mach. Learn. Res.*, 2024, 2024.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 11141–11149. AAAI Press, 2024.

Defu Cao, Furong Jia, Sercan Ö. Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Jiawei Chen and Chunhui Zhao. Addressing spatial-temporal heterogeneity: General mixed time series analysis via latent continuity recovery and alignment. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. Long sequence time-series forecasting with deep learning: A survey. *Inf. Fusion*, 97:101819, 2023.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Ping Gong, Jiawei Yi, Shengnan Wang, Juncheng Zhang, Zewen Jin, Ouxiang Zhou, Ruibo Liu, Guanbin Xu, Youhui Bai, Bowen Ye, Kun Yuan, Tong Yang, Gong Zhang, Renhai Chen, Feng Wu, and Cheng Li. HATA: trainable and hardware-efficient hash-aware top-k attention for scalable large model inference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 24856–24871. Association for Computational Linguistics, 2025.

Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. In Nafise Sadat Moosavi, Iryna Gurevych, Angela Fan, Thomas Wolf, Yufang Hou, Ana Marasovic, and Sujith Ravi (eds.), *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing, SustaiNLP@EMNLP 2021, Virtual, November 10, 2021*, pp. 39–52. Association for Computational Linguistics, 2021.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025a.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llms capabilities in time series. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b.

Yue Jiang, Yile Chen, Xiucheng Li, Qin Chao, Shuai Liu, and Gao Cong. FSTLLM: spatio-temporal LLM for few shot time series forecasting. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025.

Di Jin, Cuiying Huo, Jiayi Shi, Dongxiao He, Jianguo Wei, and Philip S. Yu. Llgformer: Learnable long-range graph transformer for traffic flow prediction. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 2860–2871. ACM, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Weiyang Kong, Ziyu Guo, and Yubao Liu. Spatio-temporal pivotal graph neural networks for traffic flow forecasting. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 8627–8635. AAAI Press, 2024.

Wenhao Li, Yuxin Zhang, Gen Luo, Haiyuan Wan, Ziyang Gong, Fei Chao, and Rongrong Ji. Spotlight attention: Towards efficient LLM generation via non-linear hashing-based KV cache retrieval. *CoRR*, abs/2508.19740, 2025.

Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. GPT-ST: generative pre-training of spatio-temporal graph neural networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and C. L. Philip Chen. Sparsetsf: Lightweight and robust time series forecasting via sparse modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(1):170–183, 2026.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *CoRR*, abs/2511.11698, 2025a.

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. In *25th IEEE International Conference on Mobile Data Management, MDM 2024, Brussels, Belgium, June 24-27, 2024*, pp. 31–40. IEEE, 2024a.

Chenxi Liu, Kethmi Hirushini Hettige, Qianxiong Xu, Cheng Long, Shili Xiang, Gao Cong, Ziyue Li, and Rui Zhao. ST-LLM+: graph enhanced spatio-temporal large language models for traffic prediction. *IEEE Trans. Knowl. Data Eng.*, 37(8):4846–4859, 2025b.

Chenxi Liu, Hao Miao, Qianxiong Xu, Shaowen Zhou, Cheng Long, Yan Zhao, Ziyue Li, and Rui Zhao. Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation. In *41st IEEE International Conference on Data Engineering, ICDE 2025, Hong Kong, May 19-23, 2025*, pp. 3165–3178. IEEE, 2025c.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 18780–18788. AAAI Press, 2025d.

Lei Liu, Shuo Yu, Runze Wang, Zhenxun Ma, and Yanming Shen. How can large language models understand spatial-temporal data? *CoRR*, abs/2401.14192, 2024b.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. CALF: aligning llms for time series forecasting via cross-modal fine-tuning. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 18915–18923. AAAI Press, 2025e.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024c.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025f.

Jiecheng Lu, Yan Sun, and Shihao Yang. In-context time series predictor. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Changxi Ma, Guowen Dai, and Jibiao Zhou. Short-term traffic flow prediction for urban road sections based on time series analysis and lstm_bilstm method. *IEEE Trans. Intell. Transp. Syst.*, 23(6): 5615–5624, 2022.

Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 19581–19588. AAAI Press, 2025.

Juntong Ni, Zewen Liu, Shiyu Wang, Ming Jin, and Wei Jin. Timedistill: Efficient long-term time series forecasting with MLP via cross-architecture distillation. *CoRR*, abs/2502.15016, 2025.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Dunwang Qin, Zhen Peng, and Lifeng Wu. Deep attention fuzzy cognitive maps for interpretable multivariate time series prediction. *Knowl. Based Syst.*, 275:110700, 2023.

Feras Saad, Jacob Burnim, Colin Carroll, Brian Patton, Urs Köster, Rif A. Saurous, and Matthew Hoffman. Scalable spatiotemporal prediction with bayesian neural fields. *CoRR*, abs/2403.07657, 2024.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, Gao Cong, Christian S. Jensen, and Xueqi Cheng. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Trans. Knowl. Data Eng.*, 37(1):291–305, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *ICLR*, 2025.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 914–921. AAAI Press, 2020.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. TEST: text prototype aligned embedding to activate llm's ability for time series. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a.

Ke Sun, Pei Liu, Pengfei Li, and Zhifang Liao. Modwavemlp: Mlp-based mode decomposition and wavelet denoising model to defeat complex structures in traffic forecasting. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 9035–9043. AAAI Press, 2024b.

Jiabin Tang, Wei Wei, Lianghao Xia, and Chao Huang. Easyst: A simple framework for spatio-temporal prediction. In Edoardo Serra and Francesca Spezzano (eds.), *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pp. 2220–2229. ACM, 2024.

Hao Wang, Jindong Han, Wei Fan, Leilei Sun, and Hao Liu. Repst: Language model empowered spatio-temporal forecasting via semantic-oriented reprogramming. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 3362–3370. ijcai.org, 2025a.

Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and Rong Jin. KVT: k-nn attention for boosting vision transformers. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXIV*, volume 13684 of *Lecture Notes in Computer Science*, pp. 285–302. Springer, 2022.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b.

Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nat. Mac. Intell.*, 5(6):602–611, 2023.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 14138–14148. AAAI Press, 2021.

Hao Xue and Flora D. Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Trans. Knowl. Data Eng.*, 36(11):6851–6864, 2024.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Changkui Yin, Yingchi Mao, Liren Deng, Meng Chen, Yi Rong, Xiaoming He, and Xiaofeng Zhou. STLLM-GAN: spatio-temporal LLM generative adversarial network for PM2.5 prediction. *Expert Syst. Appl.*, 292:128250, 2025.

Qiuhao Zeng, Jerry Huang, Peng Lu, Gezheng Xu, Boxing Chen, Charles Ling, and Boyu Wang. ZETA: leveraging z-order curves for efficient top-k attention. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a.

Michael Zhang, Simran Arora, Rahul Chalamala, Benjamin Frederick Spector, Alan Wu, Krithik Ramesh, Aaryan Singhal, and Christopher Ré. Lolcats: On low-rank linearizing of large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In Sanjoy Dasgupta, Stephan Mandt, and

Yingzhen Li (eds.), *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4222–4230. PMLR, 2024b.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 11106–11115. AAAI Press, 2021.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Xun Zhu, Yutong Xiong, Ming Wu, Gaozhen Nie, Bin Zhang, and Ziheng Yang. Weather2k: A multivariate spatio-temporal benchmark dataset for meteorological forecasting based on real-time observation data from ground weather stations. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent (eds.), *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pp. 2704–2722. PMLR, 2023.

A Linear Attention for Global Trend Modeling

To balance computational efficiency with long-range modeling capabilities when dealing with a large number of spatial nodes (i.e., large N), we use a linear attention mechanism as the global perception foundation. Its core idea is to reconstruct the standard attention computation into a linear form through a kernelized feature mapping function $\phi(\cdot)$. By leveraging the associative property of matrix multiplication, the linear attention computation is formulated as:

$$\mathrm{GLA}(Q, K, V) = \frac{\phi(Q)(\phi(K)^\top V)}{\phi(Q)(\phi(K)^\top \mathbf{1}_N)} \tag{15}$$

where $\phi(Q), \phi(K) \in \mathbb{R}^{N \times d_r}$ denote the mapped feature representations, and $\mathbf{1}_N$ is an all-ones column vector for normalization. This formulation avoids the explicit computation of the quadratic $QK^\top$ matrix, reducing the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

The effectiveness of this linearization relies on the design of the feature mapping function $\phi(\cdot)$, which is used to approximate a Gaussian kernel. Following FAVOR+ (Choromanski et al., 2021), we adopt a random feature-based approximation with Orthogonal Random Features (ORFs) to improve stability and reduce variance.

For any row vector $q \in \mathbb{R}^{1 \times d_{\text{model}}}$, the mapping function is defined as:

$$\phi(q) = \frac{1}{\sqrt{d_r}} \exp\left(q\omega - \frac{\|q\|^2}{2} \mathbf{1}_{d_r}^\top \right) \tag{16}$$

where $\omega \in \mathbb{R}^{d_{\text{model}} \times d_r}$ is constructed via QR decomposition of a Gaussian random matrix to enforce orthogonality. Compared with standard i.i.d. random features, this orthogonal design reduces the variance of kernel approximation and improves the robustness of attention computation.

B Explanations and Proofs.

B.1 Proof of Proposition 1.

Proposition 3. *For the symmetric matrix M, the matrix W containing its b leading orthonormal eigenvectors solves*

$$\max_{W} \quad \mathrm{Tr}(W^\top M W) \tag{17}$$

$$\textit{s.t.} \quad W^\top W = I. \tag{18}$$

The resulting $B = ZW$ is an approximate spectral target. In general, $W^\top W = I$ does not imply that B has zero orthogonality or balance residuals.

Proof. Because $A = ZZ^\top$ is symmetric and positive semidefinite, $M = Z_c^\top A Z_c$ is also symmetric and positive semidefinite. Let $M = V \Lambda V^\top$ be its eigendecomposition, with the eigenvalues ordered from largest to smallest. For any W satisfying $W^\top W = I$, the Rayleigh–Ritz characterization gives

$$\mathrm{Tr}(W^\top M W) \le \sum_{i=1}^{b} \lambda_i.$$

Equality is attained when the columns of W are the eigenvectors associated with the b largest eigenvalues. In the implementation, the corresponding matrix is computed from the mini-batch average of M and is detached before forming $B = ZW$.

The constraint applies to W, not to B. Indeed, $B^\top B = W^\top Z^\top Z W$ and $B^\top \mathbf{1} = W^\top Z^\top \mathbf{1}$, neither of which is generally fixed by $W^\top W = I$. Therefore, B is used as an approximate spectral target, and its normalized orthogonality and balance quantities enter Proposition 2 as residuals rather than exact constraints. This completes the proof. □

B.2 Proof of Proposition 2.

Proposition 4. *Let $\bar{B} = B/\sqrt{n}$, $\bar{H}_c = H_c/\sqrt{n}$, and $\bar{\mathbf{1}} = \mathbf{1}/\sqrt{n}$. Suppose $\|\bar{B}^\top \bar{B} - I\|_F \le \varepsilon_{\text{orth}}$, $\|\bar{B}^\top \bar{\mathbf{1}}\|_2 \le \varepsilon_{\text{bal}}$, and $\|\bar{H}_c - \bar{B}\|_F \le \delta_c$. Then*

$$\|\bar{H}_c^\top \bar{H}_c - I\|_F \le \varepsilon_{\text{orth}} + 2\delta_c\sqrt{1+\varepsilon_{\text{orth}}} + \delta_c^2$$

and

$$\|\bar{H}_c^\top \bar{\mathbf{1}}\|_2 \le \varepsilon_{\text{bal}} + \delta_c.$$

Proof. Let $\Delta_c = \bar{H}_c - \bar{B}$. For the normalized orthogonality residual,

$$\bar{H}_c^\top \bar{H}_c - I = (\bar{B}^\top \bar{B} - I) + \bar{B}^\top \Delta_c + \Delta_c^\top \bar{B} + \Delta_c^\top \Delta_c.$$

Applying the triangle inequality gives

$$\|\bar{H}_c^\top \bar{H}_c - I\|_F \le \|\bar{B}^\top \bar{B} - I\|_F + 2\|\bar{B}^\top \Delta_c\|_F + \|\Delta_c^\top \Delta_c\|_F.$$

The assumption $\|\bar{B}^\top \bar{B} - I\|_F \le \varepsilon_{\text{orth}}$ implies $\|\bar{B}\|_2 \le \sqrt{1+\varepsilon_{\text{orth}}}$. Therefore,

$$\|\bar{B}^\top \Delta_c\|_F \le \|\bar{B}\|_2 \|\Delta_c\|_F \le \delta_c\sqrt{1+\varepsilon_{\text{orth}}},$$

and $\|\Delta_c^\top \Delta_c\|_F \le \|\Delta_c\|_F^2 \le \delta_c^2$. Combining these terms proves the normalized orthogonality bound.

For the normalized balance residual,

$$\bar{H}_c^\top \bar{\mathbf{1}} = \bar{B}^\top \bar{\mathbf{1}} + \Delta_c^\top \bar{\mathbf{1}}.$$

Since $\|\bar{\mathbf{1}}\|_2 = 1$,

$$\|\bar{H}_c^\top \bar{\mathbf{1}}\|_2 \le \|\bar{B}^\top \bar{\mathbf{1}}\|_2 + \|\Delta_c\|_F \|\bar{\mathbf{1}}\|_2 \le \varepsilon_{\text{bal}} + \delta_c.$$

This completes the proof. □

Extension to hard Hamming codes. Let $L = ZW_h + \mathbf{1}b_h$, $H_c = \tanh(\beta L)$, and $H_d = \operatorname{sign}(L)$. If every logit satisfies $|l_{ij}| \ge \gamma$, then

$$|\operatorname{sign}(l_{ij}) - \tanh(\beta l_{ij})| \le 2e^{-2\beta\gamma}.$$

For a b-bit code, define $q = \|H_d - H_c\|_F$ and $\bar{q} = q/\sqrt{n}$. Then

$$\bar{q} \le 2\sqrt{b}\, e^{-2\beta\gamma}.$$

Let $\bar{H}_d = H_d/\sqrt{n}$ and $\delta_d = \|\bar{H}_d - \bar{B}\|_F$. By the triangle inequality,

$$\delta_d \le \delta_c + \bar{q}.$$

Applying the same argument as in Proposition 2 gives

$$\|\bar{H}_d^\top \bar{H}_d - I\|_F \le \varepsilon_{\text{orth}} + 2\delta_d\sqrt{1+\varepsilon_{\text{orth}}} + \delta_d^2$$

and

$$\|\bar{H}_d^\top \bar{\mathbf{1}}\|_2 \le \varepsilon_{\text{bal}} + \delta_d.$$

These bounds are margin-dependent. Under a logit perturbation ΔL, the hard codes remain unchanged only if

$$\|\Delta L\|_\infty < \min_{ij} |l_{ij}|.$$

Thus, the result transfers the residual bounds locally to hard Hamming codes whose logits remain away from the sign boundary. It does not guarantee stability of the attention output or prediction under arbitrary domain shifts.

C Construction of Differentiable Soft Top-k Weights

This section presents the detailed construction of the differentiable weight matrix $\hat{W}$ used in Soft Top-k Attention. First, we compute hash-based similarity scores:

$$F = Q_h K_h^\top, \quad Q_h = h(Q), \quad K_h = h(K) \tag{19}$$

where $h(\cdot)$ denotes the hashing function. To approximate Top-k selection, we define a dynamic threshold m, corresponding to the k-th largest value in each query row. We then compute the deviation matrix:

$$B_{dev} = F - m \tag{20}$$

A piecewise transformation is applied to construct a pre-selection weight matrix W_{mask}:

$$(W_{mask})_{ij} = \begin{cases} B_{dev,ij}/\tau, & B_{dev,ij} \geq 0 \\ (B_{dev,ij} \cdot \rho)/\tau, & B_{dev,ij} < 0 \end{cases} \tag{21}$$

where τ is a temperature parameter and $\rho > 1$ is a decay coefficient that suppresses non-top-k entries. We set $\tau = 2$ and $\rho = 3$ in all experiments. Finally, the normalized differentiable weight matrix is obtained via Softmax:

$$\hat{W} = \text{Softmax}(W_{mask}) \tag{22}$$

This formulation provides a continuous relaxation of Top-k selection, enabling gradient-based optimization while preserving sparsity structure.

D Training and Inference of the Entire Model

The training process of our model aims to efficiently transfer knowledge from a large teacher model to a lightweight student model through selective knowledge distillation. The overall optimization is guided by the joint loss function $\mathcal{L}_{\text{total}}$. In each training iteration, the main model and the learnable hash parameters W_h and b_h are jointly updated using the task, distillation, and hash approximation losses. During each forward pass, we construct the spectral matrix from the current key representation K, average it across the mini-batch, and take its leading eigenvectors as W. The eigendecomposition output W is detached before the approximate spectral target $B = KW$ is formed. Consequently, gradients do not pass through the eigendecomposition, while $\mathcal{L}_{\text{hash}}$ still updates both the learnable hash projection and the current key representation through $B = KW$.

During the inference phase, the entire teacher model is discarded, and only the student model is used for prediction. This makes the final deployed model extremely lightweight and well-suited for resource-constrained edge environments. More importantly, the inference process activates a series of computational optimizations within the hashing attention module. Specifically, the differentiable $\tanh$ function used for generating hash codes is replaced by the deterministic sign function; the similarity computation shifts from dot product to the highly efficient Hamming distance; and the differentiable Soft Top-k mechanism is switched to standard Hard Top-k selection. These transformations collectively ensure that the model is not only parameter-efficient but also achieves maximum computational speed during inference. The computational flow of our SHA is detailed in Algorithm 1, and the overall procedure for MSED-MHA is presented in Algorithm 2.

Our MHA module maintains efficient time and space complexity during inference. The hashing-based Top-k search in SHA has the complexity between $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ for N tokens, but is highly efficient in practice due to the use of Hamming distance on binary codes. Following the selection of the Top-k neighbors, attention computation is restricted to these candidates. Meanwhile, the GLA component also has linear complexity. Ignoring the Top-k search, the overall attention computation of MHA scales linearly, $\mathcal{O}(N)$. In terms of space complexity, SHA avoids storing the full $N \times N$ similarity matrix by only retaining the k neighbors for each query. As GLA also has linear space requirements, the memory usage of the MHA module grows linearly with the sequence length, i.e., $\mathcal{O}(N)$. The scalability in both computation and memory makes our model highly efficient for practical deployment.

Algorithm 1 Spectral Hashing Attention (SHA)

Input: Query $\mathbf{Q}$, Key $\mathbf{K}$, Value $\mathbf{V}$.
Parameters: Hash network parameters $\mathbf{W}_h, \mathbf{b}_h$.

Training Phase:
Compute the approximate spectral target $\mathbf{B} = \mathbf{KW}$ using the leading eigenvectors $\mathbf{W}$ of the batch-averaged spectral matrix.
Compute learnable codes $\mathbf{H_Q} \leftarrow \tanh(\beta \mathbf{L_Q})$ and $\mathbf{H_K} \leftarrow \tanh(\beta \mathbf{L_K})$.
Compute soft selection matrix $\hat{\mathbf{W}}$ via differentiable Top-k.
$\mathbf{K}' \leftarrow \hat{\mathbf{W}}\mathbf{K}$, $\mathbf{V}' \leftarrow \hat{\mathbf{W}}\mathbf{V}$.
$\mathbf{O}_{\text{SHA}} \leftarrow \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}')$.
return $\mathbf{O}_{\text{SHA}}$, $\mathcal{L}_{\text{hash}}$

Inference Phase:
Compute discrete codes $\mathbf{H_Q} \leftarrow \text{sign}(\dots)$ and $\mathbf{H_K} \leftarrow \text{sign}(\dots)$.
Select Top-k indices based on Hamming distance.
Construct sparsified $\mathbf{K}'$ and $\mathbf{V}'$ by selecting entries from $\mathbf{K}$ and $\mathbf{V}$ according to the indices.
$\mathbf{O}_{\text{SHA}} \leftarrow \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}')$.
return $\mathbf{O}_{\text{SHA}}$

Algorithm 2 Training and Inference Process of MSED-MHA

Input: Teacher $\mathcal{T}$, Student $\mathcal{S}$, Data $(\mathbf{X}, \mathcal{Y})$, Hyperparameters λ

Part 1: Training Phase
for each training iteration **do**
  Sample a mini-batch $(\mathbf{X}, \mathcal{Y})$.
  // Get predictions and post-attention representations from both models
  $\hat{\mathbf{Y}}_{\mathcal{T}}, Z^{\mathcal{T}} \leftarrow \mathcal{T}(\mathbf{X})$
  $\hat{\mathbf{Y}}_{\mathcal{S}}, \{\hat{\mathbf{Y}}_c^{\mathcal{S}}\}, \{Z_c^{\mathcal{S}}\} \leftarrow \mathcal{S}(\mathbf{X})$

  // Compute all loss components based on paper's definitions
  $\mathcal{L}_{\text{task}} \leftarrow \ell_{\text{SL1}}(\hat{\mathbf{Y}}_{\mathcal{S}}, \mathcal{Y})$
  Compute the approximate spectral target $\mathbf{B} = \mathbf{KW}$ from the student's key matrices $\mathbf{K}$.
  $\mathcal{L}_{\text{hash}} \leftarrow \|\mathbf{H}_c - \mathbf{B}\|_F^2$
  $\mathcal{L}_{\text{pred}} \leftarrow \sum_c \mathbb{I}(\ell_c^S > \ell^T) \cdot \ell_{\text{SL1}}(\hat{\mathbf{Y}}_c^{\mathcal{S}}, \hat{\mathbf{Y}}_{\mathcal{T}})$
  $\mathcal{L}_{\text{attn}} \leftarrow \sum_c \mathbb{I}(\ell_c^S > \ell^T) \cdot \ell_{\text{SL1}}(Z_c^{\mathcal{S}}, Z^{\mathcal{T}})$

  // Compute weighted total loss and update
  $\mathcal{L}_{\text{total}} \leftarrow \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{hash}}\mathcal{L}_{\text{hash}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{attn}}\mathcal{L}_{\text{attn}}$
  Update Student model $\mathcal{S}$ using $\mathcal{L}_{\text{total}}$.
end for

Part 2: Inference Phase
Discard Teacher $\mathcal{T}$.
Use lightweight Student model $\mathcal{S}$ for prediction.
return Prediction $\hat{\mathbf{Y}}_{\mathcal{S}}$

E Dataset Details and Experimental Settings

E.1 Dataset Details

This appendix provides detailed information on the public benchmark datasets used in our experiments.

- **PEMS03, PEMS04, PEMS07, PEMS08:** These four datasets contain traffic flow data from California highways, collected in real-time every 30 seconds by the California Performance Measurement System (PeMS) (Song et al., 2020) and aggregated into 5-minute intervals. The datasets record the geographical information of sensor stations, with PEMS03, PEMS04, PEMS07, and PEMS08 being datasets from four specific regions.
- **SZ_TAXI and NYC_TAXI**[2]: These are commonly used benchmark datasets in the field of spatio-temporal traffic forecasting. The former covers Shenzhen, China, for January 2015, with a 15-minute sampling interval, containing traffic speed data from 156 regions (dimension: 156, 2976, 1). The latter covers New York City, USA, from January to March 2020, with a 60-minute sampling interval, containing traffic inflow/outflow data for 263 regions (dimension: 263, 2184, 2). Both are suitable for model training and performance evaluation.
- **METR-LA:** The dataset includes traffic information collected by loop detectors on highways in Los Angeles County from March 1, 2012, to June 30, 2012. It contains data from 207 sensors, with samples taken every 5 minutes (Yi et al., 2023).
- **Finance datasets:** The finance data were obtained using the Python 'yfinance' library (based on the public Yahoo Finance API), covering the period from August 2012 to February 2024.
- **weather**[3]: This dataset was recorded every 10 minutes for the entire year of 2020 and contains 21 meteorological indicators, such as air temperature, humidity, etc.
- **WTH**[4]: This dataset contains local climatological data from nearly 1,600 locations in the United States, spanning from 2010 to 2013 (four years), collected at an hourly level. Each data point includes the target variable "wet-bulb temperature" and 11 additional climatic features. The dataset is divided into a training set (28 months), a validation set (10 months), and a test set (10 months).
- **weather2K** (Zhu et al., 2023): A Multivariate Spatio-Temporal Benchmark Dataset for Meteorological Forecasting Based on Real-Time Observation Data from Ground Weather Stations. The shape of the numpy file of weather2K-R is (1866, 13, 13632), which corresponds to 1,866 ground weather stations, 3 constants for position information, 10 meteorological factors, and 13,632 time steps with a 3-hour time resolution (Time coverage: January 1, 2017 - August 31, 2021). For the weather2K dataset, we selected three features: Air Temperature, Relative Humidity, and Wind Speed.

To address the strong correlation issue in stock data (i.e., the significant correlation between CSI300 and NASDAQ100, where knowledge of one index's future trend could influence predictions for the other), this study adopted a cross-split method to construct the training and test sets. Specifically, using the year 2020 as a boundary, we divided the financial datasets into two subsets: Finance_1 (2012–2020) and Finance_2 (2021–2024). This setup is intended to reduce information leakage between the two highly correlated datasets, thereby allowing for a more objective evaluation of the model's generalization ability in stock prediction scenarios.

E.2 Experimental Settings

Table 4: Summary of pretrain and target datasets across different domains.

Domains	Pretrain Datasets	Target Datasets
Traffic	PEMS03, PEMS04, PEMS07, NYC_TAXI	SZ_TAXI
		PEMS08
		METR-LA
Finance	Nikkei225, SP100, NASDAQ100	CSI300
		HSI
Weather	weather, WTH	weather2k

[2] https://github.com/LibCity/Bigscity-LibCity-Datasets
[3] https://www.bgc-jena.mpg.de/wetter/
[4] https://www.ncei.noaa.gov/data/local-climatological-data/

MSED-MHA is implemented using Python and PyTorch 2.2.0, and trained on an NVIDIA GeForce RTX 4090 GPU. We use Mean Absolute Error (MAE) and Mean Squared Error (MSE) as our primary evaluation metrics. In the experiments, the input window size T is set to 12, 24, or 48, and the learning rate is set to 0.01. The tanh steepness parameter is $\beta = 10$, corresponding to `alpha` in the implementation; the implementation variable `beta`=3 corresponds to the Soft Top-k decay coefficient ρ, and $\tau = 2$. The loss weights are selected on the source validation set and supported by the sensitivity analysis in Appendix G.2: $\lambda_{\text{task}} = 1$, $\lambda_{\text{hash}} = 0.1$, $\lambda_{\text{pred}} = 0.1$, and $\lambda_{\text{attn}} = 0.01$.

All models follow a unified experimental setup: the prediction horizon P is uniformly set to a value from $\{12, 24, 36, 48\}$ for all datasets. Furthermore, all models use a consistent pre-training approach: no normalization is applied during the data preprocessing stage. Instead, normalization is performed before metric calculation using statistics from the prediction set (this normalization information is not used during training and is solely for metric computation). We also applied a logarithmic transformation to the financial stock data, primarily to eliminate scale differences between stock prices, mitigate heteroscedasticity, and stabilize variance.

Due to the current lack of a standardized testing and splitting methodology for zero-shot scenarios, this study employs a consistent training strategy to ensure comparability under different experimental conditions. Our many-to-one protocol evaluates within-domain cross-dataset zero-shot transfer: a model is pretrained on multiple source datasets from one domain and evaluated directly on an unseen target dataset from the same domain, without target fine-tuning. As shown in Table 4, when predicting SZ_TAXI, the model is pretrained on PEMS03, PEMS04, PEMS07, and NYC_TAXI. Except for foundation models that cannot be trained, the source pretraining sets for all trainable models are split into 80% for pretraining and 20% for validation. For all participating models, the entire target dataset is used only for testing. This protocol does not evaluate universal transfer across domains with different variable semantics or spatial structures.

F Additional Ablation Studies

F.1 Effect of Selective Distillation

To isolate whether the teacher provides useful information beyond the student architecture, we introduce w/o-Distill, which retains the complete student but removes both prediction and post-attention representation distillation. Table 5 shows that selective distillation is better in 23 of the 24 traffic metric-horizon comparisons. Averaged over the three targets and four horizons, it reduces MSE by 19.9% and MAE by 13.6% relative to w/o-Distill. Together with the degradation of w/o-Selective in Table 2, these results indicate that teacher supervision is useful in the evaluated settings when it is applied conditionally rather than at every scale.

Table 5: Selective distillation compared with no distillation on traffic targets. The best result in each comparison is in bold.

Methods		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
MSED-MHA	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	**0.2582**	**0.2787**	**0.3092**	**0.3984**	**0.6217**	**0.8436**	**1.0927**
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	**0.3486**	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
w/o-Distill	MSE	0.2734	0.3726	0.4993	0.6479	**0.2401**	0.2744	0.3273	0.4025	0.4711	0.7356	1.0977	1.5239
	MAE	0.2574	0.3074	0.3622	0.4185	0.3340	0.3549	0.3874	0.4304	0.3640	0.4574	0.5599	0.6589

F.2 Spectral Hash Learning Compared with LSH

Rep-SHA(LSH) replaces spectral hash learning with fixed Gaussian random-hyperplane hashing while retaining the same backbone, hash length, Top-k retrieval, optimizer, learning rate, epochs, and data splits. Table 6 shows that LSH remains competitive in several MSE settings, confirming that local hash-based retrieval itself is useful. Spectral SHA is better in 14 of the 24 metric-horizon comparisons and, averaged over the three datasets and four horizons, reduces MSE from 0.4774 to 0.4584 (4.0%) and MAE from 0.3981 to 0.3522 (11.5%). For Rep-SHA(Trad.), the penalty coefficients for the quantization, balance, and orthogonality constraints are tuned on the source validation set, and the best configuration is reported.

Table 6: Spectral SHA compared with fixed random-hyperplane LSH on traffic targets. The best result in each comparison is in bold.

Methods		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
MSED-MHA	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	0.2582	0.2787	0.3092	**0.3984**	0.6217	0.8436	1.0927
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	0.3486	0.3599	0.3779	**0.2982**	**0.3804**	**0.4586**	**0.5385**
Rep-SHA(LSH)	MSE	0.3323	0.4215	0.5157	0.6182	**0.2422**	**0.2571**	**0.2663**	**0.2789**	0.4408	**0.5933**	**0.7791**	**0.9838**
	MAE	0.3165	0.3648	0.4105	0.4551	0.3323	**0.3456**	**0.3525**	**0.3592**	0.3671	0.4240	0.4904	0.5588

F.3 Ablation Study on Teacher Models

Table 7: Comparative performance of MSED-MHA and variants. The best results are in bold and the second best are underlined.

Teachers		GPT-2		Qwen		Moirai	
		MSE	MAE	MSE	MAE	MSE	MAE
PEMS08	12	**0.1966**	**0.1988**	0.3108	0.2856	0.1979	0.2091
	24	**0.3069**	**0.2562**	0.4294	0.3411	0.3207	0.2718
	36	**0.4230**	**0.3123**	0.5919	0.4069	0.4768	0.3421
	48	**0.5243**	**0.3652**	0.7762	0.4740	0.6681	0.4148
SZ_TAXI	12	**0.2474**	**0.3319**	0.2479	0.3429	0.2598	0.3412
	24	**0.2582**	**0.3486**	0.2826	0.3614	0.3218	0.3816
	36	**0.2787**	**0.3599**	0.3556	0.4068	0.4030	0.4292
	48	**0.3092**	**0.3779**	0.4544	0.4646	0.5229	0.4918
METR-LA	12	0.3984	**0.2982**	0.4781	0.3682	**0.3710**	0.3021
	24	0.6217	**0.3804**	0.7115	0.4632	**0.6182**	0.4013
	36	**0.8436**	**0.4586**	1.0067	0.5672	0.9074	0.5037
	48	**1.0927**	**0.5385**	1.3257	0.6629	1.2271	0.6027

To evaluate the robustness and versatility of our distillation framework with respect to the teacher model, we designed two additional variants for our ablation study: Qwen and Moirai. In these variants, we replaced the default teacher model with two other state-of-the-art large models, Qwen2.5-0.5B and Moirai2.0, respectively, to assess the generalization ability of our framework.

The results of this ablation study (Table 7) demonstrate that even when the teacher model is replaced with the architecturally and functionally different Qwen2.5-0.5B and Moirai2.0, the student model's performance remains at a high level. This fully demonstrates that MSED-MHA is a robust distillation framework that does not depend on a specific teacher and can efficiently extract knowledge from a diverse range of large models.

F.4 Ablation Study on Core Components

Table 8: Comparative performance of MSED-MHA and variants. The best results are in bold and the second best are underlined.

Methods		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
MSED-MHA	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	**0.2582**	**0.2787**	0.3092	0.3984	**0.6217**	0.8436	1.0927
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	0.3319	**0.3486**	**0.3599**	0.3779	0.2982	0.3804	**0.4586**	**0.5385**
w/o-SHA&MSED	MSE	0.2080	0.3228	0.4441	0.5872	**0.2448**	0.2794	0.2976	**0.3075**	0.4080	0.6635	0.9387	1.2248
	MAE	0.2093	0.2652	0.3213	0.3796	**0.3297**	0.3541	0.3665	**0.3731**	0.3105	0.4001	0.4884	0.5781
w/o-GLA&MSED	MSE	0.2595	0.3990	0.5356	0.6644	0.2969	0.3426	0.3791	0.4072	0.4688	0.6720	**0.8424**	**0.9692**
	MAE	0.2389	0.3046	0.3651	0.4210	0.3704	0.3997	0.4213	0.4372	0.3581	0.4414	0.5093	0.5593
w/o-SHA&GLA	MSE	0.2123	0.3570	0.5049	0.6532	0.2905	0.3640	0.4391	0.5385	**0.3758**	0.6283	0.8835	1.1344
	MAE	0.2020	0.2680	0.3306	0.3899	0.3538	0.3952	0.4323	0.4800	**0.2855**	**0.3780**	0.4653	0.5490

We conduct ablations using the same modular variants: (1) w/o-SHA&MSED, retaining only Linear Attention; (2) w/o-GLA&MSED, retaining only Spectral Hashing Attention; and (3) w/o-SHA&GLA, removing both attention components while keeping multi-scale distillation. The results are reported in Table 8.

The results show that each module contributes to performance gains at different horizons. Linear Attention primarily improves long-horizon predictions by capturing global trends, while Spectral Hashing Attention refines short-term, high-frequency signals. Multi-scale distillation consistently

enhances learning across all horizons, and its combination with the attention modules yields the most robust performance. These findings indicate that the observed long-horizon advantage arises from the complementary interaction of the three components rather than from a single module.

F.5 Comparison with Exact Top-k Selection

Table 9: Comparative performance of MSED-MHA and Exact Top-k. The best results are in bold.

Methods		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
MSED-MHA	MSE	**0.1966**	**0.3069**	0.4230	0.5243	**0.2474**	**0.2582**	**0.2787**	**0.3092**	0.3984	0.6217	**0.8436**	1.0927
	MAE	0.1988	0.2562	0.3123	0.3652	0.3319	**0.3486**	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
Rep-Exact	MSE	0.1978	0.3087	**0.4175**	**0.5145**	0.2488	0.2699	0.2922	0.3344	**0.3900**	**0.6104**	0.8656	**1.0202**
	MAE	**0.1980**	**0.2554**	**0.3099**	**0.3614**	**0.3311**	0.3509	0.3656	0.3918	0.3065	0.3846	0.4627	0.5468

Table 10: Computational efficiency comparison of MSED-MHA and Exact Top-k. The best results are in bold.

Methods		PEMS08	SZ_TAXI	METR-LA
MSED-MHA	Inference	**81.39**	**8.92**	**85.72**
	Parameter	**13.51M**	**13.51M**	**13.51M**
Rep-Exact	Inference	101.06	14.05	151.35
	Parameter	13.55M	13.55M	13.55M

To further evaluate the effectiveness of our approximate Top-k strategy, we design a variant named Rep-Exact, which replaces the hashing-based neighbor selection with an exact Top-k computation based on pairwise similarity.

In our method, Top-k neighbors are selected based on the similarity of learned hash codes. Specifically, we first obtain hash representations via the learnable function $h(\cdot)$. During training, $h(\cdot)$ is guided to approximate the spectral solution B, enabling efficient neighbor retrieval without explicit pairwise similarity computation. In contrast, Rep-Exact directly computes similarities between queries and keys to select the exact Top-k neighbors, which introduces higher computational overhead.

The performance comparison between the approximate Top-k and Rep-Exact is reported in Table 9. The results show that the approximate strategy achieves comparable performance to the exact Top-k variant across different settings. We further report the computational efficiency comparison in Table 10. The results indicate that the approximate Top-k significantly reduces computational cost compared to the exact approach.

Overall, these results suggest that the proposed hashing-based approximation provides a favorable trade-off between efficiency and performance.

F.6 Effect of Channel-Independent Modeling

Table 11: Comparative performance of MSED-MHA and Rep-CD. The best results are in bold.

Methods		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
MSED-MHA	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	**0.2582**	**0.2787**	**0.3092**	**0.3984**	**0.6217**	**0.8436**	**1.0927**
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	**0.3486**	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
Rep-CD	MSE	0.2947	0.4727	0.7183	1.0307	**0.2418**	0.2841	0.3570	0.4714	0.4060	0.7008	1.0385	1.3376
	MAE	0.2537	0.3266	0.4066	0.4900	0.3320	0.3587	0.4048	0.4702	0.3188	0.4382	0.5582	0.6616

To investigate the effect of the Channel-Independent (CI) modeling strategy, we further design a channel-dependent variant, denoted as Rep-CD, where all feature channels (i.e., the third dimension) are processed jointly instead of being modeled independently.

The results in Table 11 show that the CI strategy consistently achieves better performance across the evaluated spatio-temporal datasets. This may be attributed to the increased difficulty of jointly modeling temporal, spatial, and channel dependencies in a fully coupled manner, which can introduce additional optimization complexity and potentially hinder effective learning. In contrast, the CI strategy provides a more structured and stable modeling paradigm.

Therefore, Channel-Independent modeling serves as a simple yet effective design choice in our framework, which is also consistent with prior works such as Sundial (Liu et al., 2025f) and Time-MoE (Shi et al., 2025).

G Parameter Analysis.

G.1 The k Value of Top-k.

Table 12: Experiment on the impact of k on traffic datasets. The best results are in bold and the second best are underlined.

Parameter		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
k=N/100	MSE	0.2905	0.4327	0.6425	0.6425	0.2607	0.3570	0.5402	0.7953	0.4763	0.7763	1.1853	1.6369
	MAE	0.2630	0.3260	0.4034	0.4807	0.3406	0.4030	0.5063	0.6256	0.3768	0.4897	0.6147	0.7316
k=N/30	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	**0.2474**	**0.2582**	**0.2787**	**0.3092**	**0.3984**	**0.6217**	**0.8436**	**1.0927**
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	**0.3486**	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
k=N/3	MSE	0.3170	0.3990	0.5032	0.6212	0.2572	0.2647	0.2871	0.3249	0.4672	0.6521	0.8939	1.1728
	MAE	0.3080	0.3489	0.3943	0.4421	0.3563	0.3561	0.3675	0.3894	0.3591	0.4296	0.5113	0.5915

We conducted a sensitivity analysis on the value of k in our Top-k attention, with the results presented in Table 12. For each target dataset, k is recomputed as $\lfloor N/30 \rfloor$, preserving an approximately 3.3% retrieval ratio; source-domain adjacency and neighbor identities are not reused. Among the evaluated ratios, $N/30$ gives the best overall results. A smaller value such as $N/100$ can be excessively sparse, whereas a larger value such as $N/3$ can introduce less relevant nodes. The ratio remains an empirical design choice rather than a universally optimal rule.

G.2 Sensitivity Analysis of Loss Weights λ_{HASH}, λ_{ATTN}, and λ_{PRED}.

Table 13: Experiment on the impact of λ_{hash} on traffic datasets. The best results are in bold and the second best are underlined.

Parameter		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
λ_{hash}=0.01	MSE	0.3021	0.3955	0.5161	0.6587	**0.2420**	0.2634	0.2993	0.3620	0.4488	0.6420	0.8925	1.1777
	MAE	0.2875	0.3333	0.3859	0.4418	0.3392	0.3503	0.3717	0.4087	0.3537	0.4313	0.5176	0.6024
λ_{hash}=0.1	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	**0.2582**	**0.2787**	**0.3092**	**0.3984**	**0.6217**	**0.8436**	**1.0927**
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	**0.3486**	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
λ_{hash}=1	MSE	0.2089	0.3244	0.4842	0.6668	0.2483	0.2956	0.3648	0.4501	0.4289	0.7326	1.1112	1.4993
	MAE	0.2174	0.2761	0.3469	0.4157	0.3355	0.3660	0.4072	0.4537	0.3197	0.4212	0.5296	0.6273

Table 14: Experiment on the impact of λ_{attn} on traffic datasets. The best results are in bold and the second best are underlined.

Parameter		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
λ_{attn}=0.01	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	**0.2582**	**0.2787**	**0.3092**	**0.3984**	**0.6217**	**0.8436**	**1.0927**
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	**0.3319**	**0.3486**	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
λ_{attn}=0.1	MSE	0.2975	0.3927	0.5134	0.6583	0.2435	0.2698	0.3167	0.3972	0.4564	0.6660	0.9418	1.2762
	MAE	0.2691	0.3174	0.3704	0.4254	0.3380	0.3529	0.3817	0.4281	0.3582	0.4386	0.5276	0.6170
λ_{attn}=1	MSE	0.3041	0.4334	0.5893	0.7515	**0.2434**	0.2930	0.3613	0.4410	0.4421	0.6782	0.9498	1.2232
	MAE	0.2842	0.3442	0.4096	0.4720	**0.3319**	0.3635	0.4070	0.4544	0.3571	0.4551	0.5532	0.6395

We conduct a sensitivity analysis on the loss weights λ_{hash}, λ_{attn}, and λ_{pred}, with results reported in Tables 13–15. These experiments identify the selected configuration among the tested values.

Among the tested values, the best performance is achieved when $\lambda_{\text{hash}} = 0.1$, $\lambda_{\text{attn}} = 0.01$, and $\lambda_{\text{pred}} = 0.1$. This suggests that a moderate constraint on hash learning, together with relatively lighter post-attention representation alignment and balanced prediction distillation, provides an effective trade-off between structural regularization and task optimization.

In contrast, excessively small weights may lead to insufficient guidance, while overly large weights can over-constrain the model and hinder its ability to learn the primary forecasting task. Therefore, the selected configuration achieves a suitable balance between auxiliary supervision and core task learning.

Table 15: Experiment on the impact of λ_{pred} on traffic datasets. The best results are in bold and the second best are underlined.

Parameter		PEMS08				SZ_TAXI				METR-LA			
		12	24	36	48	12	24	36	48	12	24	36	48
λ_{pred}=0.01	MSE	0.2872	0.3903	0.5249	0.6805	0.2364	0.2633	0.3031	0.3625	0.4462	0.6770	0.9661	1.2748
	MAE	0.2725	0.3224	0.3802	0.4398	0.3284	**0.3456**	0.3707	0.4065	0.3586	0.4434	0.5359	0.6234
λ_{pred}=0.1	MSE	**0.1966**	**0.3069**	**0.4230**	**0.5243**	0.2474	**0.2582**	**0.2787**	**0.3092**	0.3984	**0.6217**	**0.8436**	**1.0927**
	MAE	**0.1988**	**0.2562**	**0.3123**	**0.3652**	0.3319	0.3486	**0.3599**	**0.3779**	**0.2982**	**0.3804**	**0.4586**	**0.5385**
λ_{pred}=1	MSE	0.2339	0.3437	0.4921	0.6729	**0.2342**	0.2769	0.3388	0.4324	**0.3944**	0.6271	0.9218	1.2675
	MAE	0.2393	0.2943	0.3596	0.4275	**0.3270**	0.3544	0.3925	0.4449	0.3239	0.4143	0.5134	0.6123

H Additional Efficiency Analysis

In this section, we provide a detailed analysis of the computational efficiency of MSED-MHA from both memory usage and inference complexity perspectives.

H.1 Memory Usage Comparison

We first compare the memory consumption of different methods under the same experimental settings. All experiments are conducted with the same batch size (batch size = 2), and the results are summarized in Table 16.

The results show that MSED-MHA consistently requires less memory than most baseline methods, especially LLM-based approaches such as ST-LLM+, TimeCMA, and FSCA. For example, on the PEMS08 dataset, our method uses 335MB of memory, compared to 1123MB for ST-LLM+ under the same setting. Similar trends can be observed across other datasets.

Table 16: Memory usage comparison of different methods (batch size = 2). The best results are in bold and the second best are underlined.

Methods	PEMS08	METR-LA	HSI_2	weather2k
MSED-MHA	**335mb**	**335mb**	754mb	**299mb**
TimeFilter	443mb	443mb	**746mb**	432mb
RePST	857mb	857mb	1134mb	1198mb
TimeCMA	1619mb	1619mb	1184mb	1778mb
FSCA	1117mb	1117mb	2306mb	992mb
ST-LLM+	1123mb	1123mb	1102mb	1032mb

H.2 Inference Time Complexity

Table 17: Inference time complexity comparison of different methods.

Methods	Time Complexity
MSED-MHA	$O(M \cdot C \cdot Nkd \cdot T)$
TimeFilter	$O\left(M \cdot \frac{N^2T^2}{P^2} \cdot d + M \cdot L_g \cdot \frac{NT}{P} \cdot k \cdot d\right)$
RePST	$O\left(M \cdot \frac{L_p N^2 T^2 d}{P^2} + M \cdot \frac{NTkd}{P^2} + M \cdot N^2 T\right)$
TimeCMA	$O\left(M \cdot L_{llm} \cdot N \cdot G^2 \cdot d_{llm} + M \cdot L_{ts} \cdot N^2 \cdot d + M \cdot L_{dec} \cdot N^2 \cdot d_{llm}\right)$
FSCA	$O\left(M \cdot L_{llm} \cdot \left(\frac{T}{P} + NG\right)^2 \cdot d_{llm} + M \cdot L_g \cdot \frac{T}{P} \cdot d_{llm}^2\right)$
ST-LLM+	$O\left(M \cdot \left(F \cdot N^2 + U \cdot N \cdot k\right) \cdot d + M \cdot L_{llm} \cdot N \cdot d^2\right)$

Notations. N denotes the number of spatial nodes, T the temporal length, and M the feature dimension. d and d_{llm} represent the hidden sizes of the student model and LLM, respectively. k is the Top-k value (typically $k \ll N$), and C is the number of temporal scales. L_g denotes the number of GNN layers, P the patch length, L_{llm} the number of PLM layers, L_{dec} the number of decoder layers, and L_{ts} the number of time-series encoder layers. G represents the number of prompt tokens

per variable. For ST-LLM+, F and U denote the numbers of frozen layers and graph attention layers, respectively.

We further analyze the inference time complexity of different methods, as summarized in Table 17.

Our method achieves an overall complexity of $O(M \cdot C \cdot Nkd \cdot T)$, which scales approximately linearly with respect to the number of spatial nodes N (since $k \ll N$). In contrast, most existing methods involve quadratic terms (e.g., $O(N^2)$), leading to significantly higher computational cost as N increases.

Although the absolute inference time on smaller datasets (e.g., PEMS08) is slightly higher than some lightweight baselines such as ST-LLM+, it still satisfies practical requirements. For instance, the per-sample latency is approximately $81.39/17856 \approx 0.0046$ seconds, which is suitable for real-time scenarios.

More importantly, the advantage of MSED-MHA becomes more evident on datasets with a larger number of variables. On higher-dimensional datasets such as weather2k, our method demonstrates better scalability, maintaining both efficiency and stable performance.

I Multi-Seed Results on Traffic Targets

We repeat the traffic zero-shot experiments using four seeds (55555, 66666, 88888, and 99999) while keeping the data splits and hyperparameters unchanged. Tables 18–20 report mean and standard deviation. MSED-MHA retains strong performance across the four runs, although the rankings in closely contested individual cells should not be overstated.

Table 18: Four-seed results on PEMS08. Each entry reports mean ± standard deviation; lower is better.

Method	Metric	12	24	36	48
MSED-MHA	MSE	0.2040 ± 0.0044	0.3116 ± 0.0062	0.4131 ± 0.0044	0.5296 ± 0.0057
	MAE	0.2086 ± 0.0040	0.2647 ± 0.0053	0.3164 ± 0.0057	0.3700 ± 0.0054
RePST	MSE	0.5630 ± 0.0220	0.7299 ± 0.0188	0.9571 ± 0.0370	1.2281 ± 0.0688
	MAE	0.4600 ± 0.0092	0.5310 ± 0.0134	0.6242 ± 0.0203	0.7241 ± 0.0271
iTransformer	MSE	0.2462 ± 0.0075	0.3940 ± 0.0141	0.5462 ± 0.0186	0.6973 ± 0.0243
	MAE	0.2399 ± 0.0044	0.3108 ± 0.0061	0.3775 ± 0.0071	0.4390 ± 0.0081
WPMixer	MSE	0.5257 ± 0.0549	0.8103 ± 0.0530	1.0652 ± 0.0589	1.2864 ± 0.0910
	MAE	0.3372 ± 0.0248	0.4103 ± 0.0233	0.4765 ± 0.0203	0.5362 ± 0.0178
TimeFilter	MSE	0.2042 ± 0.0054	0.3529 ± 0.0144	0.5254 ± 0.0275	0.7250 ± 0.0495
	MAE	0.2119 ± 0.0045	0.2907 ± 0.0085	0.3685 ± 0.0128	0.4456 ± 0.0184
FSCA	MSE	0.2008 ± 0.0033	0.3378 ± 0.0076	0.4767 ± 0.0125	0.6143 ± 0.0156
	MAE	0.2116 ± 0.0054	0.2779 ± 0.0051	0.3407 ± 0.0049	0.4000 ± 0.0049
TimeCMA	MSE	0.2697 ± 0.0104	0.3755 ± 0.0046	0.4754 ± 0.0028	0.5785 ± 0.0056
	MAE	0.3078 ± 0.0131	0.3670 ± 0.0092	0.4260 ± 0.0082	0.4846 ± 0.0071
ST-LLM+	MSE	0.2791 ± 0.0027	0.3915 ± 0.0059	0.4960 ± 0.0092	0.5897 ± 0.0086
	MAE	0.3102 ± 0.0040	0.3862 ± 0.0056	0.4524 ± 0.0067	0.5087 ± 0.0062

Table 19: Four-seed results on SZ_TAXI. Each entry reports mean ± standard deviation; lower is better.

Method	Metric	12	24	36	48
MSED-MHA	MSE	0.2510 ± 0.0063	0.2882 ± 0.0066	0.3106 ± 0.0096	0.3487 ± 0.0272
	MAE	0.3360 ± 0.0055	0.3628 ± 0.0053	0.3785 ± 0.0057	0.4006 ± 0.0150
RePST	MSE	0.3841 ± 0.0255	0.4232 ± 0.0270	0.4425 ± 0.0263	0.4565 ± 0.0249
	MAE	0.4310 ± 0.0173	0.4523 ± 0.0166	0.4627 ± 0.0154	0.4705 ± 0.0140
iTransformer	MSE	0.3240 ± 0.0110	0.4163 ± 0.0256	0.5002 ± 0.0353	0.5837 ± 0.0478
	MAE	0.3841 ± 0.0073	0.4412 ± 0.0154	0.4892 ± 0.0202	0.5323 ± 0.0255
WPMixer	MSE	0.5633 ± 0.0447	0.7520 ± 0.0714	0.8881 ± 0.1002	1.0064 ± 0.1395
	MAE	0.4921 ± 0.0215	0.5680 ± 0.0273	0.6192 ± 0.0334	0.6613 ± 0.0428
TimeFilter	MSE	0.2818 ± 0.0075	0.3641 ± 0.0190	0.4722 ± 0.0360	0.6291 ± 0.0641
	MAE	0.3517 ± 0.0054	0.4062 ± 0.0122	0.4692 ± 0.0202	0.5478 ± 0.0310
FSCA	MSE	0.2562 ± 0.0049	0.3261 ± 0.0094	0.4005 ± 0.0143	0.4887 ± 0.0189
	MAE	0.3323 ± 0.0023	0.3735 ± 0.0050	0.4138 ± 0.0073	0.4608 ± 0.0092
TimeCMA	MSE	0.7268 ± 0.0210	0.7803 ± 0.0254	0.7937 ± 0.0226	0.7972 ± 0.0212
	MAE	0.5940 ± 0.0153	0.6171 ± 0.0157	0.6234 ± 0.0137	0.6238 ± 0.0121
ST-LLM+	MSE	0.7707 ± 0.0157	0.8665 ± 0.0233	0.9448 ± 0.0290	1.0332 ± 0.0345
	MAE	0.5946 ± 0.0056	0.6384 ± 0.0086	0.6714 ± 0.0107	0.7047 ± 0.0119

Table 20: Four-seed results on METR-LA. Each entry reports mean ± standard deviation; lower is better.

Method	Metric	12	24	36	48
MSED-MHA	MSE	0.3859 ± 0.0160	0.6055 ± 0.0328	0.8203 ± 0.0432	1.0562 ± 0.0419
	MAE	0.2955 ± 0.0028	0.3780 ± 0.0041	0.4557 ± 0.0055	0.5367 ± 0.0091
RePST	MSE	0.5009 ± 0.0397	0.6837 ± 0.0374	0.8338 ± 0.0353	0.9556 ± 0.0311
	MAE	0.4571 ± 0.0240	0.5352 ± 0.0194	0.6075 ± 0.0159	0.6708 ± 0.0119
iTransformer	MSE	0.4565 ± 0.0104	0.7166 ± 0.0213	0.9734 ± 0.0264	1.2040 ± 0.0365
	MAE	0.3608 ± 0.0081	0.4657 ± 0.0114	0.5591 ± 0.0126	0.6369 ± 0.0145
WPMixer	MSE	0.8546 ± 0.0903	1.3402 ± 0.0898	1.7698 ± 0.1006	2.1128 ± 0.1787
	MAE	0.4716 ± 0.0307	0.5947 ± 0.0275	0.6922 ± 0.0209	0.7673 ± 0.0192
TimeFilter	MSE	0.3827 ± 0.0103	0.6641 ± 0.0212	0.9878 ± 0.0295	1.3463 ± 0.0495
	MAE	0.3073 ± 0.0086	0.4298 ± 0.0124	0.5472 ± 0.0154	0.6588 ± 0.0207
FSCA	MSE	0.3367 ± 0.0047	0.5337 ± 0.0138	0.7216 ± 0.0246	0.8935 ± 0.0307
	MAE	0.2828 ± 0.0021	0.3708 ± 0.0033	0.4512 ± 0.0045	0.5239 ± 0.0046
TimeCMA	MSE	0.3784 ± 0.0079	0.5406 ± 0.0121	0.6740 ± 0.0212	0.7836 ± 0.0300
	MAE	0.3435 ± 0.0074	0.4064 ± 0.0048	0.4566 ± 0.0057	0.4980 ± 0.0076
ST-LLM+	MSE	0.3932 ± 0.0029	0.6334 ± 0.0492	0.8054 ± 0.0194	0.9931 ± 0.0295
	MAE	0.3382 ± 0.0063	0.4582 ± 0.0110	0.5621 ± 0.0126	0.6512 ± 0.0129