

# LightSAE: Parameter-Efficient and Heterogeneity-Aware Embedding for IoT Multivariate Time Series Forecasting

Yi Ren and Xinjie Yu, *Senior Member, IEEE*

**Abstract**—Modern Internet of Things (IoT) systems generate massive, heterogeneous multivariate time series data. Accurate Multivariate Time Series Forecasting (MTSF) of such data is critical for numerous applications. However, existing methods almost universally employ a shared embedding layer that processes all channels identically, creating a representational bottleneck that obscures valuable channel-specific information. To address this challenge, we introduce a Shared-Auxiliary Embedding (SAE) framework that decomposes the embedding into a shared base component capturing common patterns and channel-specific auxiliary components modeling unique deviations. Within this decomposition, we empirically observe that the auxiliary components tend to exhibit low-rank and clustering characteristics, a structural pattern that is significantly less apparent when using purely independent embeddings. Consequently, we design LightSAE, a parameter-efficient embedding module that operationalizes these observed characteristics through low-rank factorization and a shared, gated component pool. Extensive experiments across 9 IoT-related datasets and 4 backbone architectures demonstrate LightSAE’s effectiveness, achieving MSE improvements of up to 22.8% with only 4.0% parameter increase. Code is available at <https://github.com/EDM314/LightSAE>.

**Index Terms**—Multivariate time series forecasting, channel heterogeneity, embedding mechanisms, parameter efficiency, deep learning.

## I. INTRODUCTION

THE widespread adoption of Internet of Things (IoT) systems has generated massive volumes of multivariate time series data from diverse sensors and applications across various domains. Accurately forecasting such IoT-generated data has become crucial for optimizing system performance and enabling intelligent decision-making in applications such as energy load prediction and traffic flow management [1], [2]. This growing demand has driven substantial research advances in deep learning architectures for Multivariate Time Series Forecasting (MTSF), leading to significant improvements in prediction accuracy [3].

However, a significant challenge in IoT-driven MTSF is channel heterogeneity: different channels often originate from distinct sensor types, measure different physical phenomena, or exhibit unique temporal dynamics and statistical distributions [4], [5]. As illustrated in Fig. 1(a), channels can display dramatically different patterns, from irregular fluctuations to

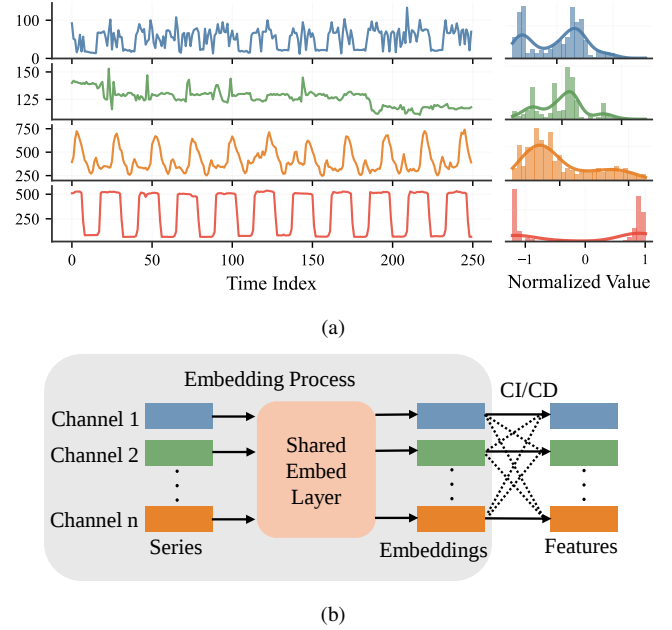


Fig. 1. Illustration of our motivation. (a) Channel heterogeneity in the Electricity dataset [6]. Four representative channels demonstrate distinct temporal patterns and statistical distributions: irregular fluctuations (top), trend with periodic spikes (second), regular oscillations (third), and square-wave patterns (bottom). (b) Existing MTSF methods uniformly adopt a shared embedding layer.

periodic oscillations, with corresponding variations in their underlying distributions. This heterogeneity suggests that channels may require specialized representational treatment to capture their unique characteristics effectively.

Despite this clear need for channel-specific modeling, existing MTSF methods have largely overlooked this challenge. Both channel-independent (CI) [7], [8] and channel-dependent (CD) [9] paradigms almost universally adopt a shared embedding layer (Fig. 1(b)), forcing heterogeneous channels through an identical transformation. This shared embedding strategy is conceptually analogous to using a single shared encoder to embed different data modalities (e.g., image and text). In fields like multimodal learning [10], [11], using modality-specific encoders [12], [13] is a well-established and effective practice at preventing the loss of unique information and improving model performance. The underlying reason is that a shared encoder, optimized for an average case, can obscure modality-specific signals, creating an informational bottleneck. However, this design has been largely under-

Y. Ren and X. Yu are with the Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: ren-y20@mails.tsinghua.edu.cn; yuxj@tsinghua.edu.cn).

Manuscript received [DATE]; revised [DATE].

explored in MTSF for handling channel heterogeneity.

This shared embedding limitation can be particularly impactful because it occurs at the initial stage of representation learning. When the initial embedding conflates channel-specific signals, valuable information risks being lost, resulting in compromised representations. Subsequent layers, regardless of their architectural sophistication, are limited to operating on these representations that lack channel-specific information. For instance, even advanced CD methods like iTransformer [9], which attempt to model intricate cross-channel interactions in deeper layers, are still constrained by the compromised channel representations produced by their shared embedding layers.

To address this challenge, we propose the Shared-Auxiliary Embedding (SAE) framework, which decomposes the embedding into a shared base component that captures global patterns common to all channels, and channel-specific auxiliary components that learn tailored representations for each channel's unique characteristics. However, directly implementing SAE faces a significant scalability challenge: when the number of channels is large, the parameter cost grows substantially due to the separate auxiliary components for each channel.

Fortunately, through deeper analysis of the SAE framework, we observe that the learned auxiliary component weights exhibit structural patterns that offer a pathway to address this scalability challenge: specifically, **low-rank and clustering characteristics**. The low-rank characteristic suggests that channel-specific deviations from common patterns can be compactly represented, while the clustering characteristic indicates that similar channels can share representational components. Notably, these characteristics are more clearly observed when auxiliary components are learned in conjunction with a shared base, and are weaker in purely independent channel embeddings without our SAE decomposition.

Building on this observation, we design LightSAE, a parameter-efficient embedding module that operationalizes these observed structures to achieve effective channel heterogeneity modeling with minimal parameter overhead. Our LightSAE addresses SAE's scalability challenge through two synergistic mechanisms: (1) low-rank factorization to compactly represent channel-specific deviations while preserving their expressiveness, and (2) a shared pool of components with a gating mechanism that facilitates both the reuse of common patterns among similar channels and the selection of specialized patterns for distinct ones, thereby leveraging the natural clustering of heterogeneous behaviors. The resulting LightSAE module is a parameter-efficient, plug-and-play component that effectively captures channel heterogeneity with substantially reduced parameter cost.

Our main contributions can be summarized as follows:

- We introduce the Shared-Auxiliary Embedding (SAE) framework and, through its analysis, observe that channel-specific auxiliary components, when disentangled from common patterns, tend to exhibit low-rank and clustering characteristics.
- We design LightSAE, a parameter-efficient embedding module, which operationalizes these observed character-

istics through low-rank factorization and a shared, gated component pool.

- We empirically validate LightSAE's effectiveness across 9 IoT-related datasets spanning diverse domains and 4 backbone architectures, achieving up to 22.8% MSE improvement with only 4.0% parameter increase.
- We conduct comprehensive ablation studies to dissect LightSAE's mechanisms, highlighting the role of the shared base in making these structures more apparent and the synergistic benefits of leveraging both low-rank and clustering observations. Furthermore, our experiments suggest that modeling channel heterogeneity at the initial embedding stage is more effective than applying it at later stages.

## II. RELATED WORK

### A. Deep Learning-based Time Series Forecasting

Deep learning approaches for MTSF are broadly categorized into channel-independent (CI) and channel-dependent (CD) methods [14]. CI models such as DLinear [15], RLinear [8], and PatchTST [7] treat each channel as a separate univariate problem with shared parameters; CD models like iTransformer [9], TimesNet [16], and TSMixer [17] explicitly model cross-channel interactions. Despite architectural differences, both paradigms almost universally adopt a single shared embedding layer, creating an initial representational bottleneck that can obscure valuable channel-specific information. Our work directly targets this bottleneck by rethinking the embedding layer itself as a channel-specific transformation.

### B. Early-Stage Heterogeneity Modeling

Our focus on embedding-stage heterogeneity is motivated by a concept well-established in other domains. In fields like multimodal learning [10], [11] and meteorological forecasting [12], [18], it is standard practice to use specialized encoders for heterogeneous data sources (e.g., image vs. text, or temperature vs. pressure) to preserve unique statistical characteristics and prevent information loss. This concept of early-stage, source-specific encoding, however, has been largely overlooked in the general MTSF literature, where different channels are typically treated uniformly by a shared embedding layer. Our work addresses this gap by analyzing this issue and proposing LightSAE, which brings this established design into MTSF in a parameter-efficient manner.

### C. Modeling Channel Heterogeneity

We contrast three lines of work with our approach.

1) *Identity and Spatio-Temporal Embeddings*: Methods such as STID [19], STAEformer [20], and GAFformer [21] address identity indistinguishability by augmenting a shared series embedding with additive positional or identity-based terms. In contrast, LightSAE models heterogeneity in the transformation function itself by learning a unique mapping for each channel. Our design is explicitly driven by two structural observations we identify, low-rankness and clusterability, to achieve parameter efficiency. These identity-based embeddings are complementary to our series-transformation approach, not substitutes.

2) *Heterogeneity in Deep Layers*: Architectures like SageFormer [2] and iTransformer [9] focus on modeling deep feature interactions across channels. These methods typically operate on representations generated by a shared embedding layer and employ sophisticated mechanisms (e.g., GNNs or attention across variates) within their deep encoder blocks to capture cross-channel dependencies. Their primary contribution lies in the interaction module, not in the initial representation learning of individual series. Our work addresses a different, upstream problem: we focus on modeling heterogeneity at the initial embedding stage itself, allowing downstream feature interaction modules to operate more effectively. The two directions are therefore highly complementary.

3) *Parameter-Efficient and Expert-Based Models*: C-LoRA [22] applies channel-wise low-rank adaptation on post-embedding features, while MoE-style approaches such as MoLE [23] and VE [24] select among experts at the output layer. Compared to these, LightSAE differs in two primary aspects: (i) Architectural Positioning: LightSAE is applied at the initial embedding layer to preserve channel-specific information from the outset. In contrast, other methods act as late-stage modules, operating on feature representations that have already been processed by a shared embedding function. (ii) Structural Motivation: Our design is directly motivated by the structural patterns observed in our SAE analysis. We first observe that channel-specific auxiliary components exhibit distinct low-rank and clustering characteristics. LightSAE is then purpose-built to incorporate these observations as inductive biases. While other works employ similar techniques like LoRA or gating, they lack a similar structural analysis. Crucially, as our analysis shows, these structural observations are significantly weaker in purely independent embeddings and emerge more clearly under our SAE decomposition. This finding highlights the value of our integrated framework, as it creates a favorable structural context for these parameter-efficient techniques to be most effective.

### III. REVISITING EMBEDDING MECHANISMS FOR MTSF

This section revisits embedding mechanisms for MTSF, where each channel typically corresponds to data from an individual IoT sensor. For clarity, our primary focus is on series-level embedding strategies [8], [9], [25], where the entire time series of each individual channel is first mapped to a latent representation. The underlying ideas are conceptually extendable to patch-level approaches [7], [26], [27].

#### A. General Pipeline of MTSF Models

As illustrated in Fig. 2(a), a general deep learning model for MTSF adopting such an initial per-channel embedding approach first maps the input series  $\mathbf{X} \in \mathbb{R}^{N \times L}$  to a set of channel-wise embeddings  $\{e_1, \dots, e_N\}$ , where  $e_i \in \mathbb{R}^{d_{\text{model}}}$  is the embedding for the  $i$ -th channel's series  $\mathbf{X}_{i,:} \in \mathbb{R}^{1 \times L}$  (denoted  $\mathbf{X}_i$  for brevity). This can be expressed as:

$$e_i = f_{\text{emb}}(\mathbf{X}_i; \Theta_{\text{emb}}) \quad \text{for } i = 1, \dots, N \quad (1)$$

These  $N$  embeddings are then typically processed by a backbone network (e.g., Transformers [9], MLPs [8]) to capture

complex dependencies and interactions, yielding a set of feature representations  $\mathbf{H}_{\text{feat}}$ :

$$\mathbf{H}_{\text{feat}} = \text{Backbone}(\{e_i\}_{i=1}^N; \Theta_{\text{backbone}}) \quad (2)$$

Finally, a projection head maps these features to the desired forecast horizon  $H$  for all  $N$  channels:

$$\hat{\mathbf{Y}} = \text{Head}(\mathbf{H}_{\text{feat}}; \Theta_{\text{proj}}) \in \mathbb{R}^{N \times H} \quad (3)$$

The entire model, comprising  $\Theta_{\text{emb}}$ ,  $\Theta_{\text{backbone}}$ , and  $\Theta_{\text{proj}}$ , is trained end-to-end by minimizing a loss function  $\mathcal{L}$ , typically the Mean Squared Error (MSE) between the prediction  $\hat{\mathbf{Y}}$  and the ground truth  $\mathbf{Y}$ :

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N \cdot H} \sum_{i=1}^N \sum_{j=1}^H (Y_{i,j} - \hat{Y}_{i,j})^2. \quad (4)$$

#### B. The Standard Shared Embedding Strategy

Contemporary MTSF models typically employ a **Shared Embedding** approach, as shown in Fig. 2(b), for the initial per-channel series-level embedding. Given an input  $\mathbf{X} \in \mathbb{R}^{N \times L}$  with  $N$  channels and lookback window length  $L$ , this approach uses a single shared linear transformation to embed each channel's series  $\mathbf{X}_i \in \mathbb{R}^{1 \times L}$  into a  $d_{\text{model}}$ -dimensional vector:

$$e_i^{\text{shared}} = \mathbf{X}_i \mathbf{W}_{\text{sh}} + \mathbf{b}_{\text{sh}}, \quad (5)$$

where  $\mathbf{W}_{\text{sh}} \in \mathbb{R}^{L \times d_{\text{model}}}$  and  $\mathbf{b}_{\text{sh}} \in \mathbb{R}^{d_{\text{model}}}$  are shared across all channels. For brevity, bias terms are omitted in subsequent equations unless explicitly stated. While this strategy is parameter-efficient, it overlooks significant channel heterogeneity and may obscure channel-specific information, thereby limiting model performance.

#### C. Shared-Auxiliary Embedding (SAE) Architecture

To learn channel-specific characteristics, a simple and intuitive approach is the **Independent Channel Embedding**, depicted in Fig. 2(c), which uses a separate linear transformation for each channel:

$$e_i^{\text{ind}} = \mathbf{X}_i \mathbf{W}_i, \quad (6)$$

where  $\mathbf{W}_i \in \mathbb{R}^{L \times d_{\text{model}}}$  are the weights for channel  $i$ . While this approach offers flexibility, it suffers from two limitations: (1) high parameter costs ( $N \times L \times d_{\text{model}}$  total parameters) that can significantly exceed the backbone model size when  $N$  is large, and (2) neglect of cross-channel commonalities, as each channel learns independently without leveraging shared patterns. These factors lead to high parameter overhead and overfitting risk, particularly in high-dimensional scenarios.

To address the challenge of channel heterogeneity while mitigating the limitations of both approaches above, we propose the **Shared-Auxiliary Embedding (SAE)** framework, shown in Fig. 2(d). SAE balances the trade-off between expressiveness and parameter efficiency by decomposing the embedding into two complementary components: a shared base that captures cross-channel commonalities (addressing the limitation of Independent Channel Embedding) and channel-specific

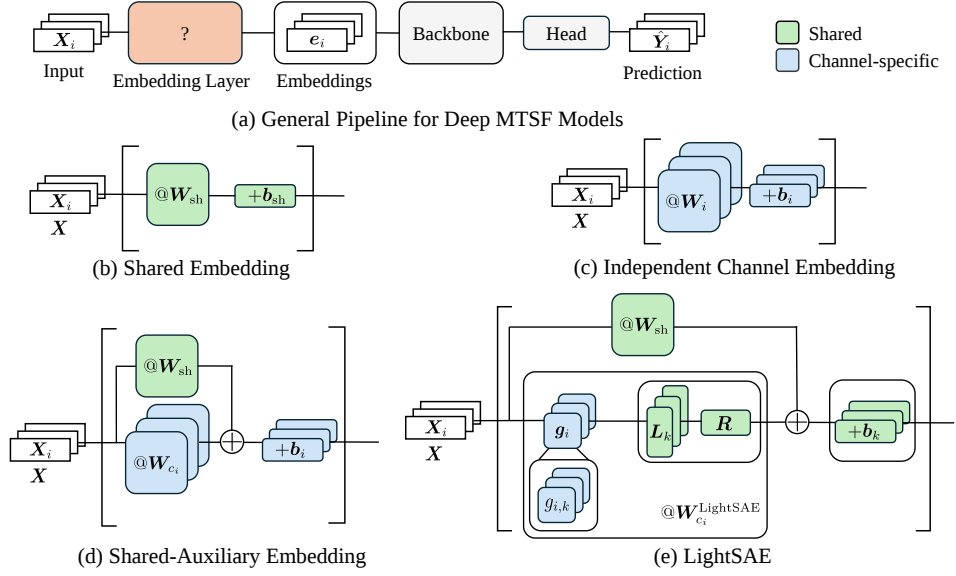


Fig. 2. A comparison of different embedding strategies for MTSF. (a) The general pipeline for deep MTSF models. (b) Standard Shared Embedding, where all channels use a single embedding layer. (c) Independent Channel Embedding, where each channel has its own separate embedding layer. (d) Our proposed Shared-Auxiliary Embedding (SAE), which combines a shared base with channel-specific auxiliary components. (e) Our final LightSAE module, which enhances SAE with parameter-efficient low-rank components and a component pool.

auxiliary components that model heterogeneous patterns (addressing the limitation of Shared Embedding):

$$e_i^{\text{SAE}} = \underbrace{X_i W_{\text{sh}}}_{\text{Shared Base}} + \underbrace{X_i W_{c_i}}_{\text{Auxiliary}}, \quad (7)$$

where  $W_{c_i} \in \mathbb{R}^{L \times d_{\text{model}}}$  are auxiliary component weight parameters for channel  $i$  (corresponding auxiliary biases  $b_{c_i}$  also exist).

While this formulation is more expressive than pure shared embedding, it still faces significant parameter scalability challenges when the number of channels is large. Fortunately, we observe that models trained with the SAE structure exhibit auxiliary weights  $W_{c_i}$  with structural characteristics, which can inform a more parameter-efficient design.

#### D. Observations on Auxiliary Component Weights under SAE

The decomposition provided by the SAE framework enables a focused analysis of the auxiliary weights  $W_{c_i}$ . A key observation of our analysis is that when disentangled from the common base component  $W_{\text{sh}}$  within our SAE decomposition, the auxiliary weights  $W_{c_i}$  tend to exhibit two structural characteristics: **low-rankness** and **clusterability**. As we will show, these structures appear to be weaker in the weights  $W_i$  from an Independent Channel Embedding, highlighting that these structural patterns become more apparent under our SAE decomposition. To the best of our knowledge, we are the first to report and leverage this structural observation in the context of MTSF.

1) *Low-Rank Characteristic of Auxiliary Component Weights*: In SAE,  $W_{\text{sh}}$  is designed to capture dominant common information, leaving  $W_{c_i}$  to model subtler, channel-specific adjustments. We hypothesize that  $W_{c_i}$ , representing

deviations from the common patterns captured by  $W_{\text{sh}}$ , can be represented in a lower-dimensional space.

Performing Singular Value Decomposition (SVD) on learned  $W_{c_i} \in \mathbb{R}^{L \times d_{\text{model}}}$  matrices from SAE, where  $W_{c_i} = U_{c_i} \Sigma_{c_i} V_{c_i}^T$  with  $\Sigma_{c_i} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$  containing singular values  $\sigma_j$  in descending order. We analyze the cumulative energy ratio  $E_k$  of the first  $k$  singular values:

$$E_k = \frac{\sum_{j=1}^k \sigma_j^2}{\sum_{j=1}^r \sigma_j^2}, \quad (8)$$

where  $r = \min(d_{\text{model}}, L)$ .

The cumulative energy analysis generally shows that  $W_{c_i}$  matrices can rapidly achieve high energy ratios (e.g., 95%) with a small number of leading components, as shown in Fig. 3 by the ‘‘Auxiliary Avg’’ curves. This indicates that  $W_{c_i}$  can be well-approximated by low-rank matrices.

In contrast, cumulative energy analysis on channel-specific weights  $W_i$  from models using Independent Channel Embedding often shows slower energy accumulation. The ‘‘Ind Avg’’ curves demonstrate this slower convergence. This suggests  $W_{\text{sh}}$  in SAE isolates the common, higher-rank components, leaving  $W_{c_i}$  to represent lower-rank channel-specific deviations.

2) *Clustering Characteristic of Auxiliary Component Weights*: We also find that auxiliary component weights  $W_{c_i}$  from SAE exhibit clustering behavior. Channels sharing similar deviation patterns from  $W_{\text{sh}}$  may have similar  $W_{c_i}$ , forming clusters.

In pairwise similarity analysis, the auxiliary weights  $W_{c_i}$  from SAE models exhibit distinct channel clusters when visualized via heatmaps. The cosine similarity between auxiliary weights of channels  $p$  and  $q$  is computed as:

$$\text{sim}(W_{c_p}, W_{c_q}) = \frac{\text{vec}(W_{c_p}) \cdot \text{vec}(W_{c_q})}{\|\text{vec}(W_{c_p})\|_2 \|\text{vec}(W_{c_q})\|_2}, \quad (9)$$

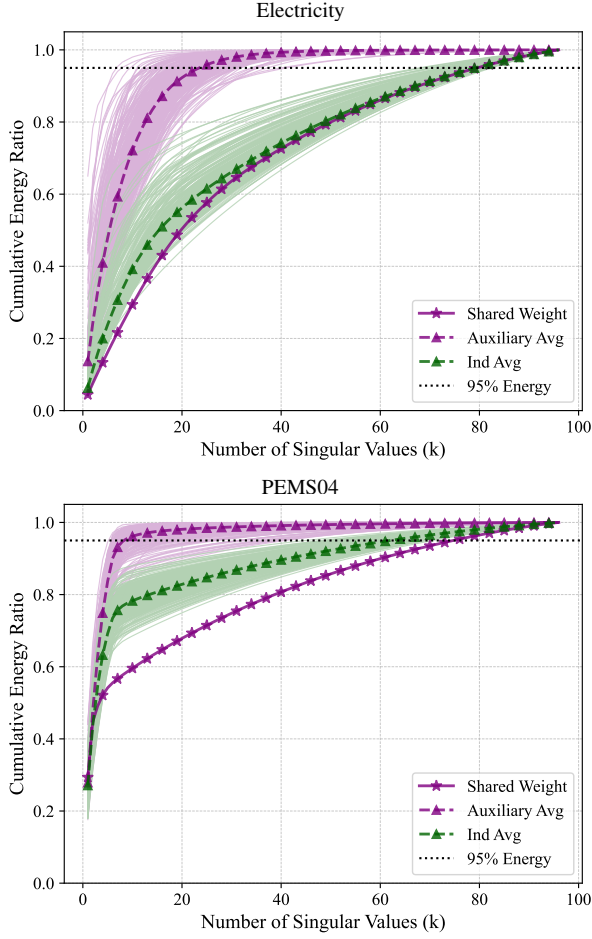


Fig. 3. Cumulative energy ratio comparison for different embedding mechanisms. “Shared Weight” represents the shared component in SAE, “Auxiliary Avg” represents the averaged value of auxiliary components  $\mathbf{W}_{c_i}$  from SAE, and “Ind Avg” represents the averaged value from Independent Channel Embedding weights,  $\mathbf{W}_i$ .

where  $\text{vec}(\cdot)$  denotes matrix vectorization. As demonstrated in Fig. 4, SAE auxiliary weights exhibit sharp contrast with distinct high similarity values close to 1 and low similarity values near 0, forming clear block-diagonal clustering structures. In contrast, Independent Channel Embedding weights show more diffuse patterns with moderate similarity values distributed across a narrower range, resulting in less pronounced clustering boundaries. This clustering suggests shared types of heterogeneity captured by  $\mathbf{W}_{c_i}$ , and SAE’s disentanglement of shared and auxiliary components makes these group structures more apparent.

### 3) Conceptual Explanation for the Observed Structures:

While a rigorous proof under realistic training dynamics is nontrivial and beyond our scope, we provide a concise, optimization-based intuition for why these structural characteristics emerge under the SAE framework. Let the per-channel losses be  $\{\mathcal{L}_i\}_{i=1}^N$  and define the average loss  $\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i$ . The gradients for the shared component ( $\mathbf{W}_{\text{sh}}$ ) and an auxiliary

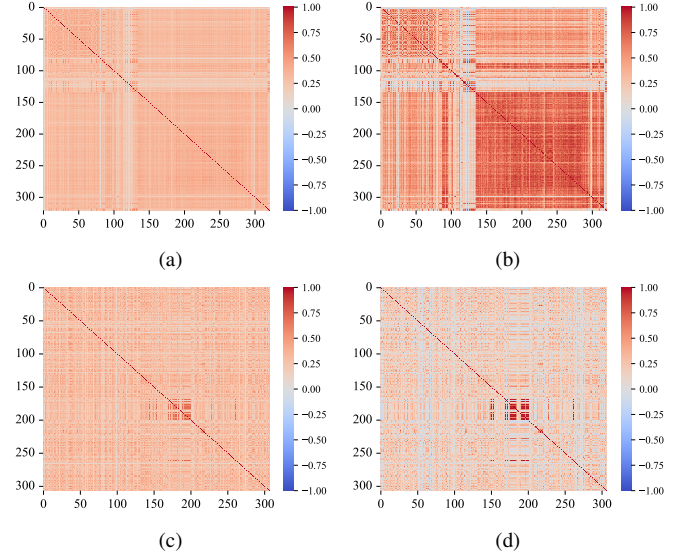


Fig. 4. Comparison of cosine similarity patterns between channel weights. (a) Similarity among Independent Channel Embedding weights  $\mathbf{W}_i$  on Electricity dataset. (b) Similarity among SAE auxiliary weights  $\mathbf{W}_{c_i}$  on Electricity dataset. (c) Similarity among Independent Channel Embedding weights  $\mathbf{W}_i$  on PEMS04 dataset. (d) Similarity among SAE auxiliary weights  $\mathbf{W}_{c_i}$  on PEMS04 dataset.

component ( $\mathbf{W}_{c_i}$ ) are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{\text{sh}}} = \frac{1}{N} \sum_{i=1}^N \frac{\partial \mathcal{L}_i}{\partial \mathbf{e}_i} \frac{\partial \mathbf{e}_i}{\partial \mathbf{W}_{\text{sh}}}, \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{c_i}} = \frac{1}{N} \frac{\partial \mathcal{L}_i}{\partial \mathbf{e}_i} \frac{\partial \mathbf{e}_i}{\partial \mathbf{W}_{c_i}}. \quad (11)$$

This formulation recovers two special cases. When  $\mathbf{W}_{c_i} = 0$ , the update reduces to the shared-only case (Eq. (5)); when  $\mathbf{W}_{\text{sh}} = 0$ , it reduces to the purely independent case (Eq. (6)).

**Rank perspective.** The observed low rank of the auxiliary weights is consistent with the optimization dynamics induced by the SAE decomposition rather than a universal guarantee. The update for  $\mathbf{W}_{\text{sh}}$  averages gradients across channels, thereby emphasizing directions repeatedly useful across the entire dataset. Consequently,  $\mathbf{W}_{\text{sh}}$  is driven to encode complex, cross-channel structures, tending to result in a higher effective rank. In contrast, each  $\mathbf{W}_{c_i}$  is driven only by the gradient of channel  $i$  and tasked with fitting the residual relative to  $\mathbf{W}_{\text{sh}}$ . This residual-fitting task is conceptually aligned with the concept of intrinsic dimensionality [28], which posits that task-specific adaptations are inherently low-dimensional [29]. This decomposition creates a structural context for the implicit bias of gradient-based optimizers to manifest. For this simpler task, the implicit bias of gradient-based optimizers towards low-rank solutions [30], [31] can manifest more strongly, guiding the learning process to find compact solution for  $\mathbf{W}_{c_i}$ . Empirically, we observe that such residuals are captured by a low-dimensional transformation, yielding a lower effective rank in  $\mathbf{W}_{c_i}$ .

In the purely independent case, however,  $\mathbf{W}_i$  must fulfill the roles of both  $\mathbf{W}_{\text{sh}}$  and  $\mathbf{W}_{c_i}$  from the SAE framework. It needs to encode both the high-rank common patterns and

the low-rank channel-specific residuals simultaneously. This entanglement of tasks results in a higher effective rank for the combined transformation. Furthermore, from an information-flow perspective, the embedding layer is the first point of data encoding. To preserve the high-rank information and maintain expressive power, the optimization process favors learning a high-rank transformation. Forcing a low-rank constraint here would risk an information bottleneck, as such a transformation is mathematically irreversible ( $\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}$ ). Accordingly,  $\mathbf{W}_i$  typically exhibits a high effective rank, consistent with Fig. 3.

**Clustering perspective.** The clustering characteristic also emerges from the SAE structure. With a shared anchor  $\mathbf{W}_{\text{sh}}$  fixing a common basis, the auxiliary weights  $\{\mathbf{W}_{c_i}\}$  represent residual deviations in the same coordinate system [32], [33], making channels with similar deviations tend to exhibit high cosine similarity, producing distinct clusters (Fig. 4b,d).

In the independent setting there is no anchor, and the learned bases  $\{\mathbf{W}_i\}$  may differ by rotations, scalings or signs; together with the entanglement of common and specific factors, this misalignment dilutes pairwise similarity and weakens cluster structure (Fig. 4a,c).

Notably, the explanation above is based on the assumption that learnable, high-rank common information exists across channels. In scenarios where such shared structure is absent, the rationale for the emergence of these low-rank and clustering patterns would be undermined.

4) *Summary of Empirical Observations under SAE:* Our empirical analysis of the SAE framework identifies two notable characteristics of the auxiliary component weights  $\mathbf{W}_{c_i}$  when disentangled from the shared base: low-rank and clustering. These structural patterns emerge more clearly under the SAE decomposition and are weaker in purely independent channel embeddings.

#### IV. LIGHTSAE: A PARAMETER-EFFICIENT HETEROGENEITY-AWARE EMBEDDING MODULE

Building on the observations from the SAE framework, our proposed LightSAE module, illustrated in Fig. 2(e), achieves parameter efficiency while effectively modeling channel heterogeneity. It accomplishes this through two synergistic mechanisms: (i) low-rank factorization to compress auxiliary components, and (ii) a shared component pool with gating to leverage clustering patterns.

##### A. Low-Rank Factorization of Auxiliary Components

The observed low-rank characteristic of auxiliary weights  $\mathbf{W}_{c_i}$  directly motivates their approximation via low-rank decomposition, similar to LoRA [29]:

$$\mathbf{W}_{c_i} \approx \mathbf{L}_{c_i} \mathbf{R}_{c_i}, \quad (12)$$

where  $\mathbf{L}_{c_i} \in \mathbb{R}^{L \times r}$  and  $\mathbf{R}_{c_i} \in \mathbb{R}^{r \times d_{\text{model}}}$ , with  $\text{rank } r \ll \min(L, d_{\text{model}})$ . This reduces parameters for each channel's auxiliary component from  $L \cdot d_{\text{model}}$  to  $r(L + d_{\text{model}})$ . However, this direct application does not leverage the clustering observation that allows different channels to benefit from similar auxiliary patterns rather than learning entirely independent ones.

##### B. Shared Pool with Gating Mechanism

To exploit the clustering observation while maintaining parameter efficiency, we introduce a shared pool of low-rank components. Instead of independent matrices  $\mathbf{L}_{c_i}, \mathbf{R}_{c_i}$  for each channel, LightSAE uses  $K$  shared left matrices  $\{\mathbf{L}_k \in \mathbb{R}^{L \times r}\}_{k=1}^K$  and a single right matrix  $\mathbf{R}_{\text{pool}} \in \mathbb{R}^{r \times d_{\text{model}}}$ .

Each channel  $i$  composes its auxiliary component by selecting from this shared pool via gating weights  $g_{i,k}$ , where  $\sum_{k=1}^K g_{i,k} = 1$ . In our implementation, we use learnable channel-specific gates: each channel has  $K$  learnable logits that are normalized via softmax to produce the gating weights. This allows channel  $i$  to form its customized auxiliary component:

$$\mathbf{W}_{c_i}^{\text{LightSAE}} = \left( \sum_{k=1}^K g_{i,k} \mathbf{L}_k \right) \mathbf{R}_{\text{pool}}. \quad (13)$$

This gating mechanism serves a dual role: it enables channels with similar residual characteristics to reuse a small set of shared components, improving parameter efficiency, while also allowing dissimilar channels to select different components, which mitigates cross-cluster interference and yields a customized transformation per channel.

The final embedding for channel  $i$  combines the shared base with the composed auxiliary component:

$$\mathbf{e}_i^{\text{LightSAE}} = \mathbf{X}_i (\mathbf{W}_{\text{sh}} + \mathbf{W}_{c_i}^{\text{LightSAE}}). \quad (14)$$

**Parameter Efficiency.** LightSAE's parameter cost is  $K \cdot L \cdot r + r \cdot d_{\text{model}} + N \cdot K$ , comprising the shared left matrices, single right matrix, and gating parameters. Given that  $r \ll \min(L, d_{\text{model}})$  and  $K \ll N$  in typical settings, this is significantly more efficient than the naive SAE approach ( $N \cdot L \cdot d_{\text{model}}$ ) when  $N$  is large, while still capturing channel heterogeneity through the observed structural patterns.

Importantly, LightSAE introduces no computational overhead during inference. Similar to LoRA, the weights can be pre-computed and merged in two steps: first, the gated auxiliary weights are combined as  $\mathbf{W}_{c_i}^{\text{aux}} = \left( \sum_{k=1}^K g_{i,k} \mathbf{L}_k \right) \mathbf{R}_{\text{pool}}$ , then merged with the shared base to form the final embedding matrix  $\mathbf{W}_i^{\text{final}} = \mathbf{W}_{\text{sh}} + \mathbf{W}_{c_i}^{\text{aux}}$  for each channel. This results in each channel having a single merged weight matrix, allowing the final embedding computation to maintain identical FLOPs to standard embedding layers.

**Plug-and-Play Integration.** LightSAE is designed to replace standard shared embedding layers in existing MTSF models. Its output preserves the expected format of channel-wise embeddings, enabling straightforward integration with various backbone architectures. The subsequent analysis demonstrates its effectiveness across diverse models and datasets.

#### V. EXPERIMENTS

To validate our proposed LightSAE, we conduct extensive experiments to evaluate four key aspects: (i) the impact of modeling channel heterogeneity on forecasting performance, (ii) the mechanisms underlying LightSAE's effectiveness and parameter efficiency, (iii) hyperparameter sensitivity and robustness across different configurations, and (iv) visualization analysis to understand the learned weights and gating mechanisms.

### A. Experimental Setup

**Datasets.** Our evaluation is conducted on 9 widely-used datasets for IoT-related MTSF. These datasets are representative and widely used in the literature [7], [9], [16] and evaluation framework TSLib [14], detailed in Table I. These IoT datasets vary significantly in their characteristics, including the number of channels (7-883), temporal resolution (5 minutes to 1 hour), and application domains.

**Backbone Models.** We integrate LightSAE with four diverse forecasting architectures:

- **RLinear** [8]: A simple yet effective linear model that directly maps input sequences to predictions.
- **RMLP** [8]: An MLP-based model with RevIN normalization that processes sequences through feedforward layers.
- **PatchTST** [7]: A Transformer model that segments time series into patches and applies self-attention.
- **iTransformer** [9]: An inverted Transformer that treats channels as tokens for multivariate modeling.

TABLE I  
DETAILED INFORMATION ABOUT THE IOT-RELATED DATASETS USED IN OUR EXPERIMENTS.

| Dataset     | Channels | Timesteps | Interval | Domain         |
|-------------|----------|-----------|----------|----------------|
| ETTh1       | 7        | 14,400    | 1 hour   | Electricity    |
| ETTh2       | 7        | 14,400    | 1 hour   | Electricity    |
| ETTm1       | 7        | 57,600    | 15 mins  | Electricity    |
| ETTm2       | 7        | 57,600    | 15 mins  | Electricity    |
| Weather     | 21       | 52,696    | 10 mins  | Weather        |
| Solar       | 137      | 52,560    | 10 mins  | Energy         |
| Electricity | 321      | 26,304    | 1 hour   | Electricity    |
| PEMS04      | 307      | 16,992    | 5 mins   | Transportation |
| PEMS07      | 883      | 28,224    | 5 mins   | Transportation |

**Implementation Details.** To ensure a fair and reproducible comparison, all experiments were conducted using TSLib [14], a widely-adopted library for MTSF evaluation [9], [34], [35]. We strictly followed its standard protocols for data splitting and normalization. For all experiments, we used a fixed prediction horizon of  $H = 96$  and evaluated across four lookback windows:  $\{96, 192, 336, 720\}$ . For all baseline models, we adopted their default hyperparameters as provided in TSLib and their original publications [8], [22], [23] without any additional tuning. For models integrated with LightSAE, we kept its own hyperparameters ( $r = 25$ ;  $K \in \{3, 7, 10\}$  depending on the dataset) consistent across all backbones and only tuned the learning rate on the validation set from  $\{1e-4, 5e-4, 1e-3, 5e-3, 1e-2\}$ .

### B. Main Results

Table II presents comprehensive forecasting results across 9 datasets and 4 backbone architectures. LightSAE achieves notable performance gains across most settings, delivering average MSE improvements of 4.7% for RLinear, 10.0% for RMLP, 8.8% for iTransformer, and 12.5% for PatchTST, with corresponding MAE improvements of 3.3%, 6.8%, 4.7%, and 8.6% respectively.

These improvements suggest effectiveness across diverse architectural paradigms: patch-based vs. variable-based embeddings, and channel-independent vs. channel-dependent models.

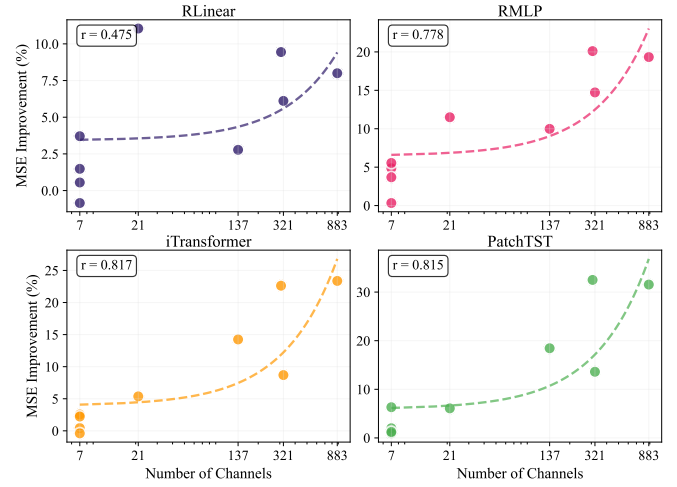


Fig. 5. Performance improvement vs. number of channels across four backbone models.

Notably, even channel-dependent models like iTransformer benefit significantly, as they still rely on shared embedding layers that constrain channel-specific representation learning from the beginning. LightSAE addresses this bottleneck by enabling heterogeneous representations right from the embedding stage.

An important observation is the correlation between dataset characteristics and improvement magnitude. As visualized in Fig. 5, performance improvements show a positive correlation with channel count across all backbone models ( $r=0.475-0.817$ ). For low-dimensional IoT systems (ETT family with 7 channels), LightSAE achieves moderate improvements (1-6%). In contrast, for large-scale IoT deployments with hundreds of channels (Electricity smart grids with 321 sensors, PEMS transportation systems with 307-883 sensors), performance gains can exceed 20% in MSE. This observation suggests that as IoT systems become more complex with additional sensors, the benefits of modeling channel heterogeneity through our proposed approach become more pronounced, likely due to increased inter-channel diversity that shared embeddings cannot adequately capture.

### C. Ablation Studies

To evaluate our core hypotheses and understand each component's contribution, we conduct a comprehensive ablation study evaluating: (i) the impact of SAE decomposition versus purely independent counterparts, and (ii) the effectiveness of different exploitation strategies for the structural observations within the SAE framework.

Table III provides the mathematical formulations for each variant, while Table IV presents an ablation analysis using the iTransformer backbone. The Configuration columns show different methods organized by embedding framework: Shared (standard shared embedding), Ind (purely independent embedding), and SAE (our proposed framework). The LR and Pool columns indicate whether each method uses Low-Rank factorization and Shared Pool components respectively.

1) *Effectiveness of SAE Decomposition:* This section validates the effectiveness of our proposed SAE decomposition

TABLE II

PERFORMANCE OF LIGHTSAE ACROSS DIFFERENT BACKBONE MODELS ON 9 DATASETS. THE PREDICTION HORIZON IS FIXED AT  $H = 96$  AND RESULTS ARE SHOWN FOR DIFFERENT LOOKBACK WINDOW SIZES  $L \in \{96, 192, 336, 720\}$ . Avg DENOTES THE AVERAGE PERFORMANCE ACROSS ALL LOOKBACK WINDOWS. **RED** INDICATES IMPROVED PERFORMANCE AND **GREEN** INDICATES DEGRADED PERFORMANCE.

| Models      | RLinear |       | w/ LightSAE |       | Improve. |        | RMLP   |       | w/ LightSAE |       | Improve. |        | iTransformer |       | w/ LightSAE |       | Improve. |        | PatchTST |       | w/ LightSAE |       | Improve. |        |        |
|-------------|---------|-------|-------------|-------|----------|--------|--------|-------|-------------|-------|----------|--------|--------------|-------|-------------|-------|----------|--------|----------|-------|-------------|-------|----------|--------|--------|
|             | Metric  | MSE   | MAE         | MSE   | MAE      | MSE    | MAE    | MSE   | MAE         | MSE   | MAE      | MSE    | MAE          | MSE   | MAE         | MSE   | MAE      | MSE    | MAE      | MSE   | MAE         | MSE   | MAE      |        |        |
| ETTh1       | 96      | 0.386 | 0.395       | 0.381 | 0.389    | 1.26%  | 1.49%  | 0.405 | 0.413       | 0.384 | 0.395    | 5.30%  | 4.24%        | 0.386 | 0.405       | 0.381 | 0.399    | 1.31%  | 1.37%    | 0.414 | 0.419       | 0.381 | 0.396    | 7.85%  | 5.48%  |
|             | 192     | 0.437 | 0.424       | 0.436 | 0.420    | 0.22%  | 0.85%  | 0.460 | 0.444       | 0.440 | 0.425    | 4.31%  | 4.20%        | 0.441 | 0.436       | 0.431 | 0.428    | 2.27%  | 1.93%    | 0.460 | 0.445       | 0.429 | 0.424    | 6.78%  | 4.61%  |
|             | 336     | 0.479 | 0.446       | 0.478 | 0.442    | 0.17%  | 0.94%  | 0.505 | 0.466       | 0.483 | 0.446    | 4.37%  | 4.31%        | 0.487 | 0.458       | 0.484 | 0.454    | 0.54%  | 0.97%    | 0.501 | 0.466       | 0.471 | 0.448    | 6.08%  | 3.89%  |
|             | 720     | 0.481 | 0.470       | 0.478 | 0.462    | 0.55%  | 1.67%  | 0.514 | 0.490       | 0.486 | 0.469    | 5.44%  | 4.36%        | 0.503 | 0.491       | 0.472 | 0.466    | 6.14%  | 5.12%    | 0.500 | 0.488       | 0.477 | 0.471    | 4.55%  | 3.43%  |
|             | Avg     | 0.446 | 0.434       | 0.443 | 0.428    | 0.55%  | 1.24%  | 0.471 | 0.453       | 0.448 | 0.434    | 4.86%  | 4.28%        | 0.454 | 0.448       | 0.442 | 0.437    | 2.56%  | 2.35%    | 0.469 | 0.455       | 0.440 | 0.435    | 6.32%  | 4.35%  |
| ETTh2       | 96      | 0.288 | 0.338       | 0.288 | 0.337    | -0.08% | 0.34%  | 0.291 | 0.342       | 0.293 | 0.343    | -0.70% | -0.42%       | 0.297 | 0.349       | 0.300 | 0.348    | -0.92% | 0.22%    | 0.302 | 0.348       | 0.294 | 0.345    | 2.48%  | 0.91%  |
|             | 192     | 0.374 | 0.390       | 0.376 | 0.392    | -0.54% | -0.59% | 0.380 | 0.396       | 0.375 | 0.392    | 1.36%  | 1.01%        | 0.380 | 0.400       | 0.378 | 0.397    | 0.59%  | 0.77%    | 0.388 | 0.400       | 0.374 | 0.395    | 3.49%  | 1.36%  |
|             | 336     | 0.415 | 0.426       | 0.418 | 0.429    | -0.83% | -0.80% | 0.419 | 0.428       | 0.416 | 0.430    | 0.55%  | -0.55%       | 0.428 | 0.432       | 0.415 | 0.429    | 3.02%  | 0.62%    | 0.426 | 0.433       | 0.417 | 0.428    | 2.22%  | 1.27%  |
|             | 720     | 0.420 | 0.440       | 0.428 | 0.442    | -1.95% | -0.35% | 0.427 | 0.443       | 0.427 | 0.441    | 0.12%  | 0.35%        | 0.427 | 0.445       | 0.431 | 0.447    | -0.96% | -0.51%   | 0.431 | 0.446       | 0.432 | 0.444    | -0.23% | 0.41%  |
|             | Avg     | 0.374 | 0.398       | 0.378 | 0.400    | -0.85% | -0.35% | 0.379 | 0.402       | 0.378 | 0.402    | 0.33%  | 0.10%        | 0.383 | 0.407       | 0.381 | 0.405    | 0.43%  | 0.28%    | 0.387 | 0.407       | 0.379 | 0.403    | 1.99%  | 0.99%  |
| ETTh1       | 96      | 0.355 | 0.376       | 0.339 | 0.367    | 4.59%  | 2.50%  | 0.337 | 0.374       | 0.315 | 0.354    | 6.45%  | 5.42%        | 0.334 | 0.376       | 0.334 | 0.369    | 0.14%  | 1.97%    | 0.329 | 0.367       | 0.321 | 0.361    | 2.28%  | 1.66%  |
|             | 192     | 0.391 | 0.392       | 0.376 | 0.383    | 3.72%  | 2.39%  | 0.379 | 0.391       | 0.358 | 0.379    | 5.41%  | 3.15%        | 0.377 | 0.392       | 0.373 | 0.389    | 1.10%  | 0.80%    | 0.367 | 0.385       | 0.365 | 0.386    | 0.61%  | -0.31% |
|             | 336     | 0.424 | 0.415       | 0.408 | 0.403    | 3.67%  | 2.88%  | 0.412 | 0.412       | 0.390 | 0.401    | 5.45%  | 2.58%        | 0.426 | 0.415       | 0.404 | 0.409    | 5.23%  | 1.43%    | 0.399 | 0.410       | 0.390 | 0.403    | 2.16%  | 1.71%  |
|             | 720     | 0.487 | 0.450       | 0.473 | 0.438    | 2.85%  | 2.78%  | 0.478 | 0.447       | 0.455 | 0.439    | 4.87%  | 1.69%        | 0.491 | 0.459       | 0.479 | 0.451    | 2.44%  | 1.69%    | 0.454 | 0.439       | 0.460 | 0.449    | -1.29% | -2.37% |
|             | Avg     | 0.414 | 0.408       | 0.399 | 0.397    | 3.71%  | 2.64%  | 0.401 | 0.406       | 0.380 | 0.393    | 5.55%  | 3.21%        | 0.407 | 0.411       | 0.397 | 0.404    | 2.23%  | 1.47%    | 0.387 | 0.400       | 0.384 | 0.400    | 0.94%  | 0.17%  |
| ETTh2       | 96      | 0.182 | 0.265       | 0.177 | 0.257    | 2.80%  | 2.87%  | 0.180 | 0.262       | 0.176 | 0.257    | 2.37%  | 2.16%        | 0.180 | 0.264       | 0.178 | 0.261    | 1.24%  | 1.01%    | 0.175 | 0.259       | 0.175 | 0.257    | 0.22%  | 0.80%  |
|             | 192     | 0.246 | 0.304       | 0.242 | 0.299    | 1.50%  | 1.58%  | 0.246 | 0.303       | 0.237 | 0.297    | 3.28%  | 2.00%        | 0.250 | 0.309       | 0.249 | 0.308    | 0.45%  | 0.18%    | 0.241 | 0.302       | 0.240 | 0.300    | 0.58%  | 0.56%  |
|             | 336     | 0.307 | 0.342       | 0.304 | 0.339    | 0.93%  | 0.81%  | 0.308 | 0.343       | 0.293 | 0.334    | 4.72%  | 2.38%        | 0.311 | 0.348       | 0.316 | 0.349    | -1.65% | -0.39%   | 0.305 | 0.343       | 0.296 | 0.336    | 2.88%  | 1.90%  |
|             | 720     | 0.407 | 0.398       | 0.404 | 0.396    | 0.69%  | 0.53%  | 0.407 | 0.398       | 0.390 | 0.391    | 4.37%  | 1.63%        | 0.412 | 0.407       | 0.419 | 0.408    | -1.62% | -0.36%   | 0.402 | 0.400       | 0.397 | 0.396    | 1.26%  | 1.06%  |
|             | Avg     | 0.285 | 0.327       | 0.282 | 0.323    | 1.48%  | 1.45%  | 0.285 | 0.327       | 0.274 | 0.320    | 3.68%  | 2.04%        | 0.288 | 0.332       | 0.290 | 0.332    | -0.39% | 0.11%    | 0.281 | 0.326       | 0.277 | 0.322    | 1.23%  | 1.08%  |
| Weather     | 96      | 0.192 | 0.232       | 0.162 | 0.206    | 15.49% | 11.41% | 0.196 | 0.235       | 0.161 | 0.207    | 17.61% | 11.92%       | 0.174 | 0.214       | 0.158 | 0.203    | 9.19%  | 5.28%    | 0.177 | 0.218       | 0.161 | 0.209    | 8.79%  | 4.14%  |
|             | 192     | 0.240 | 0.271       | 0.208 | 0.247    | 13.20% | 8.91%  | 0.240 | 0.271       | 0.207 | 0.249    | 13.56% | 8.09%        | 0.221 | 0.254       | 0.208 | 0.249    | 5.88%  | 1.98%    | 0.225 | 0.259       | 0.208 | 0.251    | 7.45%  | 3.02%  |
|             | 336     | 0.292 | 0.307       | 0.264 | 0.288    | 9.61%  | 6.23%  | 0.291 | 0.307       | 0.264 | 0.291    | 9.42%  | 5.30%        | 0.278 | 0.296       | 0.264 | 0.291    | 5.17%  | 1.61%    | 0.278 | 0.297       | 0.264 | 0.291    | 5.16%  | 1.94%  |
|             | 720     | 0.364 | 0.353       | 0.343 | 0.338    | 5.89%  | 4.19%  | 0.363 | 0.353       | 0.343 | 0.343    | 5.40%  | 2.77%        | 0.358 | 0.347       | 0.354 | 0.350    | 1.25%  | -0.90%   | 0.354 | 0.348       | 0.344 | 0.343    | 2.93%  | 1.42%  |
|             | Avg     | 0.272 | 0.291       | 0.244 | 0.270    | 11.05% | 7.69%  | 0.272 | 0.291       | 0.244 | 0.273    | 11.50% | 7.02%        | 0.258 | 0.278       | 0.246 | 0.273    | 5.37%  | 1.99%    | 0.259 | 0.280       | 0.244 | 0.274    | 6.08%  | 2.63%  |
| Solar       | 96      | 0.322 | 0.339       | 0.308 | 0.334    | 4.28%  | 1.41%  | 0.233 | 0.296       | 0.214 | 0.268    | 8.33%  | 9.34%        | 0.203 | 0.237       | 0.177 | 0.237    | 12.85% | 0.18%    | 0.234 | 0.286       | 0.206 | 0.265    | 11.79% | 7.31%  |
|             | 192     | 0.359 | 0.356       | 0.347 | 0.354    | 3.23%  | 0.53%  | 0.260 | 0.316       | 0.236 | 0.284    | 9.42%  | 10.20%       | 0.233 | 0.261       | 0.213 | 0.259    | 8.55%  | 0.93%    | 0.267 | 0.310       | 0.223 | 0.268    | 16.56% | 13.67% |
|             | 336     | 0.397 | 0.369       | 0.389 | 0.369    | 2.12%  | 0.07%  | 0.276 | 0.323       | 0.247 | 0.289    | 10.45% | 10.39%       | 0.248 | 0.273       | 0.201 | 0.274    | 18.77% | -0.24%   | 0.290 | 0.315       | 0.224 | 0.271    | 22.72% | 13.97% |
|             | 720     | 0.397 | 0.356       | 0.391 | 0.362    | 1.47%  | -1.80% | 0.273 | 0.316       | 0.241 | 0.281    | 11.74% | 11.11%       | 0.249 | 0.275       | 0.207 | 0.272    | 16.81% | 1.07%    | 0.289 | 0.317       | 0.223 | 0.272    | 22.70% | 14.11% |
|             | Avg     | 0.369 | 0.355       | 0.359 | 0.355    | 2.78%  | 0.05%  | 0.261 | 0.313       | 0.234 | 0.281    | 9.98%  | 10.26%       | 0.233 | 0.262       | 0.200 | 0.260    | 14.24% | 0.48%    | 0.270 | 0.307       | 0.219 | 0.269    | 18.45% | 12.26% |
| Electricity | 96      | 0.201 | 0.281       | 0.187 | 0.266    | 7.05%  | 5.34%  | 0.190 | 0.272       | 0.155 | 0.246    | 18.77% | 9.75%        | 0.148 | 0.240       | 0.134 | 0.231    | 9.17%  | 3.82%    | 0.181 | 0.270       | 0.149 | 0.239    | 17.77% | 11.32% |
|             | 192     | 0.201 | 0.283       | 0.187 | 0.269    | 6.73%  | 4.94%  | 0.195 | 0.279       | 0.167 | 0.256    | 14.49% | 8.21%        | 0.162 | 0.253       | 0.155 | 0.250    | 4.49%  | 1.07%    | 0.188 | 0.274       | 0.162 | 0.254    | 13.92% | 7.46%  |
|             | 336     | 0.215 | 0.298       | 0.203 | 0.284    | 5.84%  | 4.59%  | 0.212 | 0.296       | 0.184 | 0.274    | 13.04% | 7.61%        | 0.178 | 0.269       | 0.168 | 0.265    | 5.43%  | 1.44%    | 0.204 | 0.293       | 0.180 | 0.274    | 11.99% | 6.50%  |
|             | 720     | 0.257 | 0.331       | 0.244 | 0.318    | 5.02%  | 4.06%  | 0.255 | 0.331       | 0.223 | 0.308    | 12.61% | 6.97%        | 0.225 | 0.317       | 0.190 | 0.286    | 15.68% | 9.77%    | 0.246 | 0.324       | 0.220 | 0.310    | 10.73% | 4.41%  |
|             | Avg     | 0.218 | 0.298       | 0.205 | 0.284    | 6.11%  | 4.73%  | 0.213 | 0.295       | 0.182 | 0.271    | 14.73% | 8.13%        | 0.178 | 0.270       | 0.162 | 0.258    | 8.69%  | 4.02%    | 0.205 | 0.290       | 0.177 | 0.269    | 13.61% | 7.42%  |
| PEMS04      | 96      | 1.127 | 0.812       | 1.012 | 0.758    | 10.19% | 6.63%  | 0.262 | 0.358       | 0.192 | 0.298    | 26.47% | 16.70%       | 0.150 | 0.262       | 0.115 | 0.225    | 23.48% | 14.26%   | 0.291 | 0.389       | 0.166 | 0.269    | 42.93% | 30.77% |
|             | 192     | 1.150 | 0.808       | 1.038 | 0.757    | 9.77%  | 6.37%  | 0.299 | 0.384       | 0.234 | 0.329    | 21.66% | 14.29%       | 0.195 | 0.298       | 0.140 | 0.247    | 28.24% | 17.32%   | 0.303 | 0.396       | 0.203 | 0.298    | 32.91% | 24.75% |
|             | 336     | 0.838 | 0.645       | 0.764 | 0.610    | 8.84%  | 5.40%  | 0.276 | 0.363       | 0.231 | 0.327    | 16.41% | 10.08%       | 0.192 | 0.294       | 0.167 | 0.265    | 13.11% | 10.05%   | 0.284 | 0.379       | 0.208 | 0.302    | 26.87% | 20.26% |
|             | 720     | 0.977 | 0.725       | 0.890 | 0.684    | 8.97%  | 5.68%  | 0.324 | 0.399       | 0.272 | 0.362    | 15.88% | 9.24%        | 0.256 | 0.341       | 0.190 | 0.292    | 25.64% | 14.47%   | 0.324 | 0.406       | 0.236 | 0.328    | 27.28% | 19.28% |
|             | Avg     | 1.023 | 0.748       | 0.926 | 0.702    | 9.44%  | 6.02%  | 0.290 | 0.376       | 0.232 | 0.329    | 20.10% | 12.58%       | 0.198 | 0.299       | 0.153 | 0.257    | 22.62% | 14.02%   | 0.301 | 0.392       | 0.203 | 0.299    | 32.50% | 23.76% |
| PEMS07      | 96      | 1.096 | 0.795       | 1.001 | 0.741    | 8.66%  | 6.84%  | 0.306 | 0.378       | 0.236 | 0.314    | 22.82% | 16.75%       | 0.139 | 0.245       | 0.105 | 0.204    | 24.38% | 16.93%   | 0.346 | 0.404       | 0.195 | 0.277    | 43.56% | 31.39% |
|             | 192     | 1.149 | 0.798       | 1.054 | 0.744    | 8.29%  | 6.83%  | 0.368 | 0.416       | 0.284 | 0.350    | 22.80% | 15.79%       | 0.192 | 0.287       | 0.143 | 0.232    | 25.43% | 19.28%   | 0.352 | 0.409       | 0.247 | 0.306    | 29.98% | 25.15% |
|             | 336     | 0.821 | 0.631       | 0.758 | 0.596    | 7.63%  | 5.57%  | 0.325 | 0.385       | 0.272 | 0.341    | 16.36% | 11.49%       | 0.191 | 0.283       | 0.156 | 0.240    | 18.14% | 14.98%   | 0.322 | 0.388       | 0.230 | 0.305    | 28.71% | 21.29% |
|             | 720     | 0.972 | 0.710       | 0.900 | 0.668    | 7.40%  | 5.88%  | 0.382 | 0.422       | 0.323 | 0.378    | 15.35% | 10.39%       | 0.256 | 0.331       | 0.191 | 0.272    | 25.52% | 17.78%   | 0.379 | 0.423       | 0.288 | 0.341    | 23.92% | 19.54% |
|             | Avg     | 1.009 | 0.733       | 0.928 | 0.687    | 8.00%  | 6.28%  | 0.345 | 0.400       | 0.279 | 0.346    | 19.33% | 13.61%       | 0.194 | 0.286       | 0.149 | 0.237    | 23.37% | 17.24%   | 0.350 | 0.406       | 0.240 | 0.307    | 31.54% | 24.34% |
| Dataset Avg |         |       |             |       |          | 4.70%  | 3.30%  |       |             |       |          | 10.01% | 6.80%        |       |             |       |          | 8.79%  | 4.66%    |       |             |       |          | 12.52% | 8.56%  |

TABLE IV

ABLATION STUDY RESULTS WITH FRAMEWORK-BASED ANALYSIS USING ITRANSFORMER BACKBONE. THE LOOKBACK WINDOW IS FIXED AT  $L = 96$  AND RESULTS ARE AVERAGED ACROSS DIFFERENT PREDICTION HORIZONS  $H \in \{96, 192, 336, 720\}$ . WE COMPARE DIFFERENT EMBEDDING FRAMEWORKS (SHARED, IND, SAE) AND TECHNICAL COMPONENTS (LR: LOW-RANK, POOL: SHARED POOL). RESULTS SHOW MSE, PERFORMANCE IMPROVEMENT (%), PARAMETER COUNT (M), AND PARAMETER INCREASE (%).

| Configuration   |            |          |          | Solar        |               |              |              | PEMS04       |               |              |              |
|-----------------|------------|----------|----------|--------------|---------------|--------------|--------------|--------------|---------------|--------------|--------------|
| Method          | Frame.     | LR       | Pool     | MSE          | $\Delta\%$    | Params       | $\Delta\%$   | MSE          | $\Delta\%$    | Params       | $\Delta\%$   |
| Baseline        | Shared     | ×        | ×        | 0.233        | +0.0%         | 3.38M        | +0.0%        | 0.198        | +0.0%         | 4.96M        | +0.0%        |
| Ind-Full        | Ind        | ×        | ×        | 0.224        | +3.8%         | 10.13M       | +199.9%      | 0.178        | +10.2%        | 20.15M       | +306.6%      |
| Ind-LR          | Ind        | ✓        | ×        | 0.223        | +4.5%         | 5.48M        | +62.2%       | 0.176        | +11.2%        | 9.73M        | +96.3%       |
| Ind-Pool        | Ind        | ×        | ✓        | 0.220        | +5.7%         | 3.82M        | +13.1%       | 0.175        | +11.5%        | 5.40M        | +9.0%        |
| LightSAE-Ind    | Ind        | ✓        | ✓        | 0.232        | +0.7%         | 3.44M        | +1.7%        | 0.201        | −1.4%         | 5.10M        | +3.0%        |
| SAE-Full        | SAE        | ×        | ×        | 0.219        | +6.1%         | 10.18M       | +201.3%      | 0.174        | +12.4%        | 20.20M       | +307.6%      |
| SAE-LR          | SAE        | ✓        | ×        | 0.213        | +8.7%         | 5.53M        | +63.7%       | 0.162        | +18.2%        | 9.78M        | +97.3%       |
| SAE-Pool        | SAE        | ×        | ✓        | 0.211        | +9.5%         | 3.87M        | +14.6%       | 0.165        | +16.5%        | 5.45M        | +10.0%       |
| <b>LightSAE</b> | <b>SAE</b> | <b>✓</b> | <b>✓</b> | <b>0.200</b> | <b>+14.4%</b> | <b>3.49M</b> | <b>+3.2%</b> | <b>0.153</b> | <b>+22.8%</b> | <b>5.15M</b> | <b>+4.0%</b> |

performance drop occurs because the design rationale of LightSAE is violated in the Ind setting. Specifically, our structural analysis shows that low-rank and clustering patterns are observed in the auxiliary weights of SAE, but are weaker in the Ind framework. Therefore, applying these mechanisms without the shared component of SAE leads to suboptimal results for two key reasons. For Ind-LR, imposing a low-rank constraint on a weight matrix that does not empirically exhibit strong low-rankness (Fig. 3) creates an information bottleneck, as it must compress both high-rank common patterns and low-rank specific details into a single, constrained transformation. Similarly, for Ind-Pool, the component sharing mechanism is less effective because, as shown in Fig. 4, the independent weights lack the clear clustering structure needed for efficient sharing. In essence, the SAE framework is crucial because it creates the conditions under which these structural properties emerge, allowing them to be effectively exploited.

2) *Effectiveness and Efficiency of LightSAE Components Design:* Having established that structural patterns are more apparent under SAE decomposition, we now dissect how LightSAE operationalizes these observations for improved performance and efficiency. The results in the lower part of Table IV illustrate this progression within the SAE framework. The SAE-Full model suggests that modeling heterogeneity is beneficial (MSE drops from 0.198 to 0.174 on PEMS04), but at an unsustainable parameter cost (+307.6%).

Fortunately, our component-wise ablations within the SAE framework show how each observed characteristic translates into an effective inductive bias. The low-rank structure serves as a compression prior in SAE-LR, where enforcing a low-rank constraint on auxiliary weights operationalizes our structural observation. This acts as a regularizer, encouraging the model to capture only the most significant channel-specific deviations. The result is not just substantial parameter reduction (to +97.3%), but also improved performance (MSE 0.162 on PEMS04), suggesting that the low-rank structure is an important characteristic. Similarly, the clustering structure functions as a sharing prior in SAE-Pool, which operationalizes our clustering observation by enabling channels to share a common pool of

auxiliary components. This leverages the insight that channels naturally form groups with similar representational needs, allowing for both parameter reuse among similar channels and the selection of distinct components for dissimilar ones. SAE-Pool achieves competitive performance (MSE 0.165 on PEMS04) with a modest parameter budget (+10.0%), confirming the effectiveness of this structural exploitation.

Finally, LightSAE synergistically combines both structural priors by using a shared pool of low-rank components, simultaneously applying compression and sharing mechanisms. This combination yields the best accuracy (0.153 MSE, a 22.8% improvement over baseline) with minimal parameter overhead (+4.0%). This demonstrates that the low-rank and clustering observations are complementary, and that designing an architecture that explicitly respects both achieves a better trade-off between expressiveness and efficiency.

3) *Analysis of Application Point:* A design choice of LightSAE is its application as an input embedding module. To validate this choice, we compare its performance when applied at different architectural positions. Since LightSAE targets linear transformations, it can be applied to replace any linear layer in the general MTSF pipeline shown in Fig. 2(a). Specifically, we examine three configurations: (1) embedding layer only, (2) projection head (output layer) only, and (3) both layers. This analysis helps determine the optimal placement strategy for LightSAE within the model architecture.

As shown in Fig. 6, the embedding layer approach achieves the best performance on three out of four datasets, while the output layer approach generally performs worst across all datasets. This supports our hypothesis that heterogeneity-aware modeling at the embedding stage enables all subsequent layers to benefit from channel-specific representations. While applying LightSAE at both layers can yield marginal gains in some cases (e.g., PEMS04), the initial input-level application appears to be the primary driver of performance improvement. Given the added parameter cost, applying LightSAE solely at the embedding layer offers a compelling parameter-performance trade-off. In contrast, addressing heterogeneity only at the output layer may fail to inform intermediate feature learning,

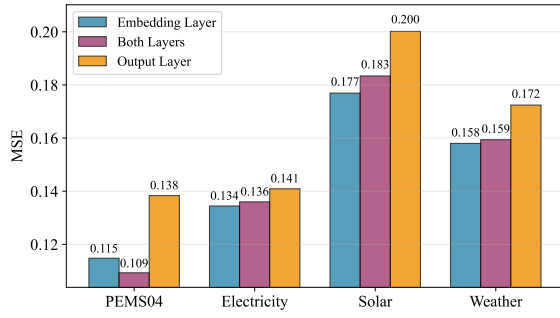


Fig. 6. Performance comparison of LightSAE applied at different positions in the model architecture (input length 96, prediction length 96).

limiting the model's ability to learn discriminative channel-specific patterns.

4) *Comparison with Related Methods*: To further validate the effectiveness of our approach, we compare LightSAE against recent methods that can also model channel heterogeneity. These methods represent different approaches to modeling channel-specific patterns:

**C-LoRA** [22] applies low-rank adaptation to individual channels after token embedding to achieve channel-specific adaptation.

**MoLE** [23] employs mixture-of-experts at the output layer with timestamp-dependent routing, training multiple linear experts where a router model adaptively weighs their outputs based on temporal periodicity.

**VE** [24] uses variate-dependent experts at the output layer, creating channel-specific representations through variate-aware expert selection to model heterogeneous channel patterns.

TABLE V

PERFORMANCE COMPARISON WITH RELATED METHODS. RESULTS SHOW MSE AND MAE VALUES WITH LOOK-BACK WINDOW 96 AND AVERAGED OVER FOUR PREDICTION HORIZONS {96, 192, 336, 720}. BEST RESULTS ARE HIGHLIGHTED IN BOLD.

| Methods      | Metric      | Baseline |       | + C-LoRA |       | + MoLE |       | + VE  |       | + LightSAE   |              |
|--------------|-------------|----------|-------|----------|-------|--------|-------|-------|-------|--------------|--------------|
|              |             | MSE      | MAE   | MSE      | MAE   | MSE    | MAE   | MSE   | MAE   | MSE          | MAE          |
| RMLP         | Weather     | 0.272    | 0.291 | 0.247    | 0.276 | 0.254  | 0.282 | 0.251 | 0.276 | <b>0.244</b> | <b>0.273</b> |
|              | Solar       | 0.261    | 0.313 | 0.245    | 0.291 | 0.246  | 0.291 | 0.241 | 0.289 | <b>0.234</b> | <b>0.281</b> |
|              | Electricity | 0.213    | 0.295 | 0.198    | 0.287 | 0.201  | 0.290 | 0.195 | 0.285 | <b>0.182</b> | <b>0.271</b> |
|              | PEMS04      | 0.290    | 0.376 | 0.240    | 0.341 | 0.245  | 0.348 | 0.243 | 0.338 | <b>0.232</b> | <b>0.329</b> |
|              | PEMS07      | 0.345    | 0.400 | 0.290    | 0.353 | 0.295  | 0.355 | 0.302 | 0.361 | <b>0.279</b> | <b>0.346</b> |
| iTransformer | Weather     | 0.258    | 0.278 | 0.251    | 0.275 | 0.249  | 0.271 | 0.253 | 0.276 | <b>0.246</b> | <b>0.273</b> |
|              | Solar       | 0.233    | 0.262 | 0.211    | 0.261 | 0.214  | 0.273 | 0.209 | 0.260 | <b>0.200</b> | <b>0.260</b> |
|              | Electricity | 0.178    | 0.270 | 0.168    | 0.265 | 0.171  | 0.268 | 0.168 | 0.266 | <b>0.162</b> | <b>0.258</b> |
|              | PEMS04      | 0.198    | 0.299 | 0.165    | 0.267 | 0.169  | 0.274 | 0.168 | 0.273 | <b>0.153</b> | <b>0.257</b> |
|              | PEMS07      | 0.194    | 0.286 | 0.160    | 0.262 | 0.165  | 0.269 | 0.163 | 0.269 | <b>0.149</b> | <b>0.237</b> |

The results in Table V provide empirical support for our design choices, demonstrating the importance of both architectural positioning and our SAE framework.

First, the consistent performance advantage of LightSAE over MoLE and VE suggests the importance of **architectural positioning**. By specializing only at the output layer, MoLE and VE are fundamentally limited, as they operate on feature representations that have already been homogenized by a shared input embedding. LightSAE's superior results suggest that to effectively model heterogeneity, intervention at the initial embedding stage is more effective for preserving and propagating channel-specific information throughout the model.

Second, the results highlight the important role of our **SAE decomposition** in providing a clear motivation for efficient heterogeneity modeling. Methods like MoLE and VE lack this decomposition entirely; consequently, their use of expert-mixing is not guided by the low-rank and clustering priors that we observe in our SAE analysis. In contrast, C-LoRA does leverage low-rank adaptation but fails to exploit the clustering structure. This leads to an inefficient design where a distinct adapter is required for every channel, overlooking the potential for sharing components among similar channels. In contrast, LightSAE synergistically leverages the low-rank and clustering characteristics from the SAE decomposition, achieving parameter efficiency and high performance.

#### D. Hyperparameter Sensitivity

We first analyze the sensitivity of LightSAE to its two main hyperparameters: rank  $r$  and pool size  $K$ .

As shown in Fig. 7, the model exhibits robust performance across different rank values. For iTransformer, performance generally improves as the rank increases from  $r = 1$ , reaching an optimal range around  $r = 25$  to  $r = 40$ , before slightly degrading at higher ranks. This suggests that a moderate rank is sufficient to capture essential channel-specific information without overfitting.

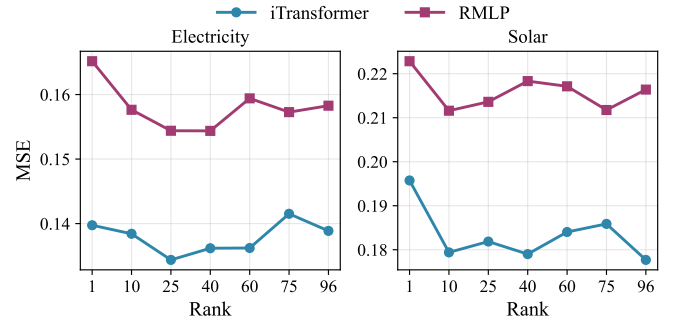


Fig. 7. Model performance under different rank values.

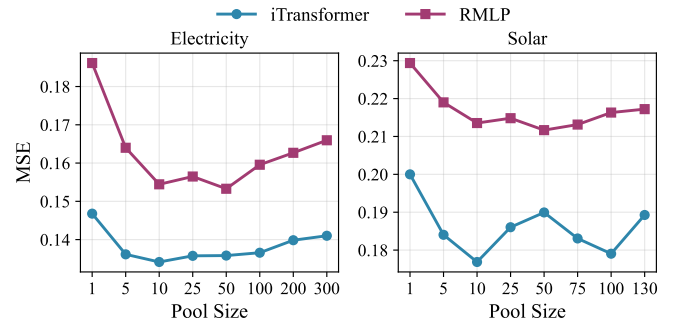


Fig. 8. Model performance under different pool size values.

Similarly, Fig. 8 illustrates the model's robustness to the pool size  $K$ . Both models show a trend where performance improves significantly when moving from  $K = 1$  to  $K = 10$ , indicating the benefit of a diverse component pool. Beyond this point, performance stabilizes, with  $K = 10$  representing

a good trade-off that balances expressiveness and parameter efficiency. The overall stability across a wide range of both  $r$  and  $K$  values suggests the robustness of our LightSAE module.

We further evaluate the robustness of LightSAE to variations in input sequence length. As shown in Fig. 9, both backbone models generally benefit from increased input lengths, with RMLP exhibiting more pronounced improvements. For iTransformer, performance improves with longer sequences but shows slight degradation at very long lengths. Overall, these results indicate that using LightSAE still enables the model to benefit from increased input sequence length.

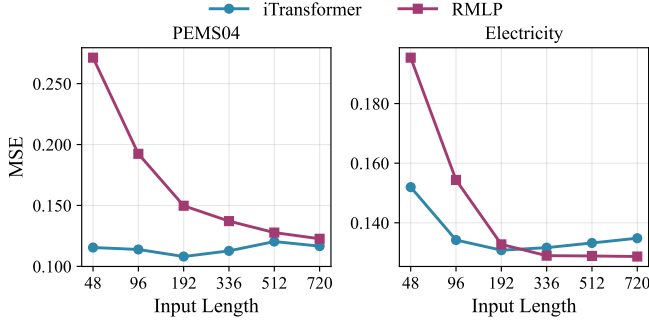


Fig. 9. Performance across different input sequence lengths.

#### E. Visualization Analysis

We conduct visualization analyses of LightSAE using the RMLP backbone on the Electricity dataset.

1) *Gating Weights Analysis*: Fig. 10 shows a t-SNE [36] projection of the  $K$ -dimensional gating weight vectors  $\{g_{i,k}\}$  for all channels. Channels with proximate positions in the embedding space exhibit similar temporal dynamics (e.g., channels 65 and 66 show nearly identical oscillatory patterns), while distant channels display substantial differences (e.g., channel 317 exhibits distinct spike patterns). This spatial-temporal correspondence suggests that LightSAE's gating mechanism captures meaningful channel relationships based on temporal characteristics. The right panel shows prediction performance on representative channels, where the baseline struggles while LightSAE closely follows the ground truth, particularly evident in channels 1 and 317.

2) *Auxiliary Component Analysis*: Fig. 11 provides a visual analysis of the learned component pool. The heatmap on the left shows the cosine similarities between component matrices  $L_k \in \mathbb{R}^{L \times r}$ . A key observation is the diversity within the pool, evidenced by the prevalence of component pairs with low similarity or near-orthogonality. This diversity suggests the pool has learned a varied set of components that can function as a basis. Such a structure is beneficial as it allows channels with similar characteristics to select and reuse a common subset of components, while enabling channels with unique patterns to compose distinct transformations from other, dissimilar components, thus allowing distinct transformations for different channel clusters while maintaining parameter efficiency. The visualizations on the right offer a qualitative confirmation, showing that components with high similarity

(e.g., 0.64 for components 0 and 1) share visual patterns, whereas orthogonal ones (e.g., 0.02 for components 2 and 5) exhibit no discernible visual correlation. At the same time, the heatmap also shows components of high similarity (e.g., over 0.9 for components 5, 6, and 9), indicating that some redundancy still exists within the learned pool. While these highly similar, but not identical components may capture subtle variations of a common pattern, this observation also points to a potential avenue for future improvement, such as incorporating an orthogonality constraint during training to further encourage diversity and reduce redundancy.

#### VI. LIMITATIONS

While LightSAE demonstrates strong empirical performance, we acknowledge several limitations that suggest avenues for future work. First, our design is empirically motivated by the observed low-rank and clustering characteristics. While we provide conceptual intuition, a formal theoretical explanation for why and under what conditions these structures emerge remains an open question. Consequently, its effectiveness may be reduced on datasets where such structural patterns are much less pronounced. Furthermore, our evaluation is focused on IoT-related benchmarks; future work could explore its generalizability to other time series domains such as finance or healthcare. Second, while our extensive experiments across nine representative datasets did not show evidence of the gating mechanism collapsing to a trivial solution, we acknowledge that such a risk could theoretically exist. However, our current framework does not include explicit regularization mechanisms (e.g., load balancing losses [37]) to formally prevent collapsing. Incorporating such regularization to enhance robustness is valuable, and we leave it for future exploration. Finally, as our approach is based on a linear additive decomposition, exploring non-linear extensions could further improve the modeling of complex heterogeneity patterns. We leave these promising directions for future exploration.

#### VII. CONCLUSION

This work challenges the prevailing one-size-fits-all embedding paradigm in MTSF by addressing the issue of channel heterogeneity, which is particularly pronounced in IoT systems with diverse sensor networks. We introduced the Shared-Auxiliary Embedding (SAE) framework, which decomposes representations into shared and auxiliary components. Within this decomposition, we empirically observed that channel-specific information, when disentangled from common patterns, exhibits low-rank and clustering structures. This observation directly motivated the design of LightSAE, a module that operationalizes these characteristics through low-rank factorization and a shared, gated component pool that promotes both parameter sharing and channel-specific specialization. Experiments across 9 IoT-related datasets and 4 backbone architectures support our approach, showing that LightSAE can deliver substantial performance gains (e.g., up to 22.8% MSE improvement with a minimal 4.0% parameter increase). As a plug-and-play module, LightSAE offers a practical and effective solution for enhancing channel-specific representation

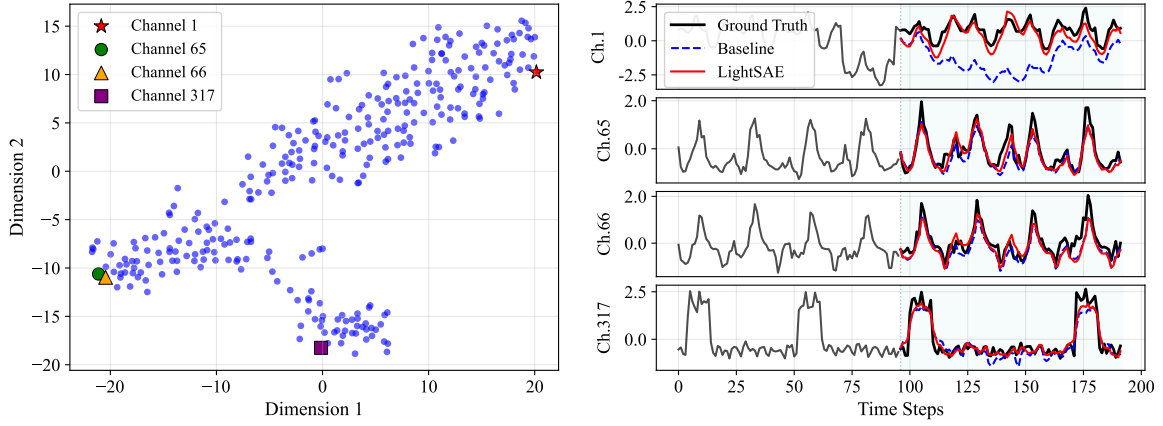


Fig. 10. Visualization of learned gating weights using t-SNE (left) and prediction performance on representative channels (right).

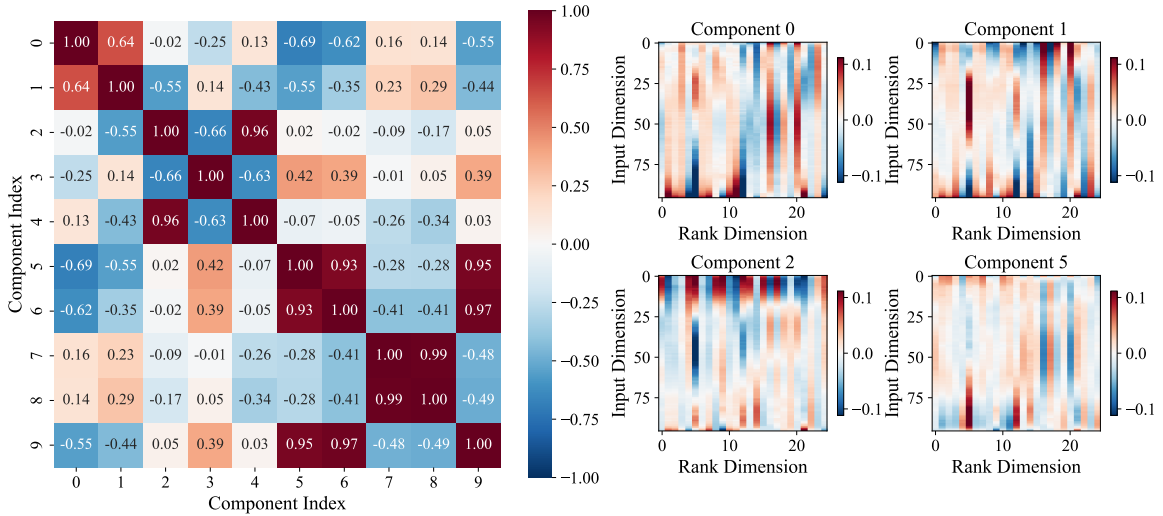


Fig. 11. Analysis of auxiliary component pool in LightSAE. (Left) Pairwise cosine similarity heatmap. (Right) Visualization of the left-side low-rank matrices,  $L_k \in \mathbb{R}^{L \times r}$ . For each component heatmap, the Y-axis represents input sequence length ( $L$ ) and the X-axis represents the rank of the low-rank matrices ( $r$ ).

learning in existing MTSF models. Given the effectiveness demonstrated across these diverse and representative datasets, we anticipate that the observed structural patterns and the proposed LightSAE framework will be beneficial for a broader range of IoT-related time series applications.

## REFERENCES

- [1] L. Shen, Y. Wang, X. Fan, Y. Wei, and H. Qiu, "Variable-Dynamic Multivariate Time-Series Forecasting for IoT Systems," *IEEE Internet of Things Journal*, vol. 12, no. 14, pp. 28 479–28 492, 2025.
- [2] Z. Zhang, L. Meng, and Y. Gu, "SageFormer: Series-Aware Framework for Long-Term Multivariate Time-Series Forecasting," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 18 435–18 448, 2024.
- [3] X. Qiu, J. Hu, L. Zhou, X. Wu, J. Du, B. Zhang, C. Guo, A. Zhou, C. S. Jensen, Z. Sheng *et al.*, "TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods," *Proceedings of the VLDB Endowment*, vol. 17, no. 9, pp. 2363–2377, 2024.
- [4] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible Instance Normalization for Accurate Time-series Forecasting against Distribution Shift," in *International conference on learning representations*, 2021.
- [5] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting," *Advances in neural information processing systems*, vol. 35, pp. 9881–9893, 2022.
- [6] A. Trindade, "ElectricityLoadDiagrams20112014," UCI Machine Learning Repository, 2015. [Online]. Available: <https://archive.ics.uci.edu/dataset/321/electricityloadDiagrams20112014>
- [7] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A Time Series is Worth 64 Words: Long-Term Forecasting with Transformers," in *The Eleventh International Conference on Learning Representations*, 2023.
- [8] Z. Li, S. Qi, Y. Li, and Z. Xu, "Revisiting Long-Term Time Series Forecasting: An Investigation on Linear Mapping," *arXiv preprint arXiv:2305.10721*, 2023.
- [9] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting," in *The Twelfth International Conference on Learning Representations*, 2024.
- [10] M. Meila and T. Zhang, Eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021.
- [11] Y. Zhang, K. Gong, K. Zhang, H. Li, Y. Qiao, W. Ouyang, and X. Yue, "Meta-transformer: A unified framework for multimodal learning," *arXiv preprint arXiv:2307.10802*, 2023.
- [12] K. Chen, T. Han, F. Ling, J. Gong, L. Bai, X. Wang, J.-J. Luo, B. Fei, W. Zhang, X. Chen *et al.*, "The operational medium-range deterministic weather forecasting can be extended beyond a 10-day lead time," *Communications Earth & Environment*, vol. 6, no. 1, p. 518, 2025.
- [13] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière *et al.*, "Gemma 3 technical report," *arXiv preprint arXiv:2503.19786*, 2025.

- [14] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep Time Series Models: A Comprehensive Survey and Benchmark," *arXiv preprint arXiv:2407.13278*, 2024.
- [15] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.
- [16] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis," in *The Eleventh International Conference on Learning Representations*, 2023.
- [17] S.-A. Chen, C.-L. Li, S. O. Arik, N. C. Yoder, and T. Pfister, "TSMixer: An all-MLP architecture for time series forecast-ing," *Transactions on Machine Learning Research*, 2023.
- [18] S. Zhu, Y. Chen, P. Yu, X. Qu, Y. Zhou, Y. Ma, Z. Zhao, Y. Liu, H. Mi, and B. Wang, "Puyun: Medium-range global weather forecasting using large kernel attention convolutional networks," *arXiv preprint arXiv:2409.02123*, 2024.
- [19] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting," in *Proceedings of the 31st ACM international conference on information & knowledge management*, 2022, pp. 4454–4458.
- [20] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-Temporal Adaptive Embedding Makes Vanilla Transformer SOTA for Traffic Forecasting," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 4125–4129.
- [21] J. Xiao, R. Liu, and E. L. Dyer, "GAFormer: Enhancing Timeseries Transformers through Group-Aware Embeddings," in *The Twelfth International Conference on Learning Representations*, 2024.
- [22] T. Nie, Y. Mei, G. Qin, J. Sun, and W. Ma, "Channel-Aware Low-Rank Adaptation in Time Series Forecasting," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 3959–3963.
- [23] R. Ni, Z. Lin, S. Wang, and G. Fanti, "Mixture-of-Linear-Experts for Long-Term Time Series Forecasting," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 4672–4680.
- [24] S. Wang, Z. Man, Z. Cao, J. Zheng, and Z. Ge, "VE: Modeling Multivariate Time Series Correlation with Variate Embedding," in *2024 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE, 2024, pp. 105–110.
- [25] Y. Wang, H. Wu, J. Dong, G. Qin, H. Zhang, Y. Liu, Y. Qiu, J. Wang, and M. Long, "TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024, pp. 469–498.
- [26] Y. Zhang and J. Yan, "Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting," in *The eleventh international conference on learning representations*, 2023.
- [27] J. Liu, C. Liu, G. Woo, Y. Wang, B. Hooi, C. Xiong, and D. Sahoo, "UniTST: Effectively Modeling Inter-Series and Intra-Series Dependencies for Multivariate Time Series Forecasting," in *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.
- [28] A. Aghajanyan, S. Gupta, and L. Zettlemoyer, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 2021, pp. 7319–7328.
- [29] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations*, 2022.
- [30] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Implicit regularization in matrix factorization," *Advances in neural information processing systems*, vol. 30, 2017.
- [31] J. Kim, J. Kim, and E. K. Ryu, "Lora training provably converges to a low-rank global minimum or it fails loudly (but it probably won't fail)," *arXiv preprint arXiv:2502.09376*, 2025.
- [32] B. N. Flury, "Common principal components in k groups," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 892–898, 1984.
- [33] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 171–184, 2012.
- [34] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik, "UNITS: A Unified Multi-task Time Series Model," *Advances in Neural Information Processing Systems*, vol. 37, pp. 140 589–140 631, 2024.
- [35] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.
- [36] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [37] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of Machine Learning Research*, vol. 23, no. 120, pp. 1–39, 2022. [Online]. Available: <http://jmlr.org/papers/v23/21-0998.html>

## VIII. BIOGRAPHY



**Yi Ren** was born in Hunan, China, in October 1998. He received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, Tsinghua University. His research interests include artificial intelligence and data mining.



**Xinjie Yu** (Senior Member, IEEE) was born in Guizhou, China, in February 1973. He received B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 2001, respectively. He is currently a professor in the Department of Electrical Engineering, Tsinghua University. His research interests include pulsed power supply, current sensors, and computational intelligence.