Enhancing Time Series Forecasting via Distribution Calibration and Prototypical Pattern Modeling

1st Tielin Yin
School of Computer Science and Technology
Changsha University of Science and Technology
Changsha, China
1524305349@qq.com

2nd Ping Li
School of Computer Science and Technology
Changsha University of Science and Technology
Changsha, China
lping9188@163.com

***Abstract*—Multivariate Time Series Forecasting (MTSF) is crucial in domains such as energy dispatch and traffic management. Recently, the inverted embedding paradigm that maps variable sequences as independent tokens has significantly enhanced models' representational capacity. However, practical applications still face two major bottlenecks: (1) feature distribution distortion caused by noise interference leads to normalization failure, preventing models from capturing stable data distributions; (2) traditional value embedding ignores behavioral pattern disparities among heterogeneous variables, resulting in blind coupling between different variables. To address these challenges, this paper proposes the AW-PVA model. First, we design the Adaptive Weighted Reversible Instance Normalization module (AW-RevIN), which dynamically identifies reliable signals through adaptive weighting to suppress outlier interference and enhance distribution fidelity; meanwhile, sparse regularization is introduced to guide the model in accurately anchoring key distributional characteristics. Second, we design the Prototypical Vector Augmentation (PVA) module, which constructs a library of typical behavior patterns through K globally shared prototypical vectors, enhancing the discrimination of heterogeneous variables while significantly improving the modeling precision of complex variable relationships. Experiments demonstrate that AW-PVA outperforms existing state-of-the-art models in long-term forecasting tasks.**

***Index Terms*—Multivariate Time Series Forecasting (MTSF), Non-stationarity, Prototype Learning.**

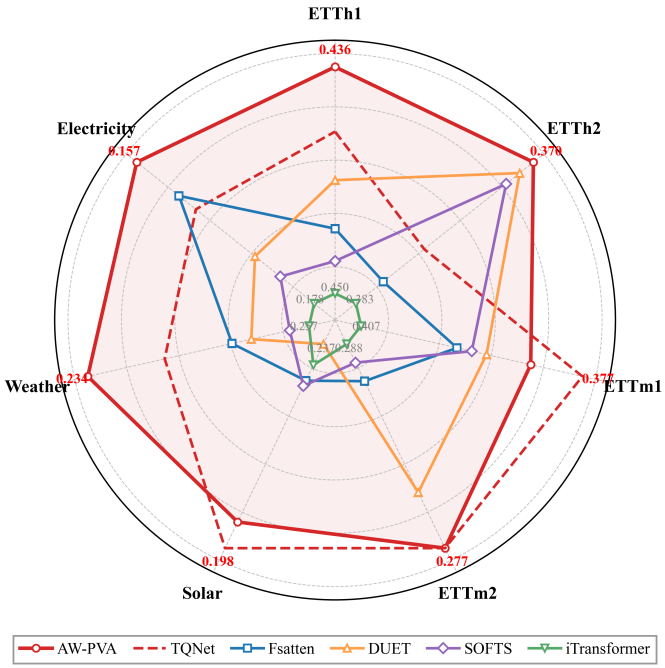

Fig. 1. Performance of AW-PVA Model. Results are presented as the average MSE.

I. Introduction

Multivariate Time Series Forecasting (MTSF) plays an indispensable role in numerous domains of modern society, including energy dispatch [1], traffic flow prediction [2], financial market analysis [3], [4], healthcare monitoring [5], and weather forecasting [6]. Early statistical models such as ARIMA [3] and autoregressive state space models [7], despite their strong interpretability, are constrained by limited parameter capacity. Subsequently emerged deep neural network models, including Recurrent Neural Networks (RNNs) [8], [9] and Convolutional Neural Networks (CNNs) [1], [10], though demonstrating excellence in capturing short-term dynamics, face challenges of gradient vanishing and computational efficiency in long-term forecasting tasks [2], [11].

In recent years, the Transformer architecture [5] and its variants [12]–[15] have demonstrated exceptional performance in MTSF tasks, leveraging the inherent advantages of self-attention mechanisms in modeling long-range dependencies. However, recently, DLinear [16], a lightweight model based on Multi-Layer Perceptron(MLP), has outperformed Transformer models and their variants on multiple benchmarks with a relatively simple linear structure. This phenomenon has prompted researchers to reconsider whether there exist modeling approaches more aligned with temporal characteristics within the Transformer architecture. In this context, iTransformer [17] introduced a novel inverted modeling paradigm that maps the entire observation sequence of each variable into an independent feature token, moving away from the conventional approach of embedding all observations within a single timestamp. This paradigm enables more explicit modeling of global spatial correlations among variables and is regarded as a highly promising direction for long-term forecasting, with its superior effectiveness validated in various subsequent studies, such as SOFTS [18] and TQNet [19].

Despite the remarkable performance achieved by these works, they still face two critical challenges when dealing with the complex and dynamic data of the real world.

First, non-stationarity in time series leads to the distortion of feature distributions. Non-stationarity refers to the phenomenon where statistical properties (mean and variance) evolve dynamically over time [20]. Although RevIN [21], Non-stationary Transformer [20], and FITS [22] attempt to calibrate distribution shifts by aligning distributions, these methods all rely on the "equal-weight contribution" assumption—that is, assuming all sampling points contribute uniformly to the global distribution. However, [23] point out that random interference and extreme outliers, which are ubiquitous in real-world observation sequences, can severely distort the estimation of statistical properties. This suggests that the "equal-weight contribution" assumption encounters a significant performance bottleneck when handling low-quality data. The inclusion of inferior samples not only fails to provide effective distributional information but also impairs the fidelity of feature representations due to statistical distortion.

Second, under the inverted embedding paradigm, models typically employ homogeneous embedding layers to process all variables, overlooking their intrinsic physical meanings or categorical attributes. This results in the blind coupling of variables with distinct evolutionary patterns [24]. Although Crossformer [25] and FourierGNN [4] attempt to enhance correlations from the spatial dimension, the challenge of how to automatically mine behavioral patterns among heterogeneous variables and achieve knowledge sharing [26], [27] remains unresolved. Existing methods struggle to distinguish between variables with starkly different characteristics (periodicity versus burstiness) [28], [29], thereby limiting the representation precision of models in MTSF tasks.

To address the aforementioned challenges, we propose the AW-PVA model. Figure 1 illustrates the performance of AW-PVA, and our main contributions are summarized as follows:

- **Adaptive Weighted Reversible Instance Normalization (AW-RevIN) Module.** This module optimizes the distribution estimation process by dynamically adjusting the weights of individual sampling points, while employing a sparsity penalty term to guide weight allocation—encouraging the model to precisely capture core sampling points most representative of distributional patterns. This mechanism ensures automatic anchoring to reliable signal regions within samples, thereby suppressing random interference and extreme outliers while maintaining distribution fidelity.
- **Prototypical Vector Augmentation (PVA) Module.** This module injects exclusive identity signatures for different variables by constructing a global knowledge base comprising K prototypical vectors. This design not only enhances the model's discriminability among heterogeneous variables but also facilitates information sharing between variables exhibiting similar behavioral patterns, thereby addressing the issue of blind coupling caused by homogenized variable embeddings.
- Across 28 forecasting tasks on 7 widely used real-world datasets, AW-PVA achieves 38 best and 16 second-best results out of 56 total evaluation metrics. This performance consistently surpasses state-of-the-art models, including iTransformer [17], SOFTS [18], and TQNet [19].Our code is available at https://github.com/1524305349/AW-PVA.

II. RELATED WORK

Time Series Forecasting. Early statistical models [3], [7] are constrained by limited parameter capacity, while subsequent RNNs [8], [9] and CNNs [1], [10], [30] face challenges of gradient vanishing and computational efficiency in long-term forecasting tasks [2], [11].In recent years, the Transformer architecture, leveraging the potential of self-attention mechanisms for sequence modeling, has inspired a wide range of variants [12]–[15], [25]. However, recent lightweight MLP-based models [16], [31] have demonstrated that simple linear structures can outperform complex attention mechanisms on certain benchmarks. This phenomenon has prompted the academic community to re-examine time-series modeling approaches. Most recently, the inverted modeling paradigm [17] has significantly enhanced the accuracy of long-term forecasting tasks by redefining the basic units of feature representation.

Non-stationarity and Normalization. Non-stationarity in time series represents a fundamental obstacle to long-term forecasting [11], [20], [21], [32]. To calibrate distribution shift, [21] proposed reversible instance normalization, which significantly improves performance by aligning statistical characteristics before and after embedding. [20] combated information loss by integrating de-stationarization factors into the self-attention mechanism, while [33] explored normalization techniques within the frequency domain. Additionally, several studies [13], [14], [16], [26] mitigated the impact of non-stationarity through seasonal-trend decomposition, whereas [22], [30] investigated multiscale interactions and sampling compression, respectively. However, the aforementioned works all rely on the assumption of "equal-weight contribution" from observation points when using normalization to estimate statistics. As highlighted in [23], random fluctuations and outliers in real-world data can severely distort the estimation of empirical mean and standard deviation. Existing methods lack discriminative mechanisms for sampling point reliability when computing distributions, rendering them prone to statistical shifts when confronted with non-stationary noise.

Channel and Prototype Learning. In multivariate modeling, the strategic trade-off between Channel Independence (CI) and Channel Mixing (CM) has emerged as a central focus of recent research [34]. Models adopting the CI strategy [16], [24], [26] enhance modeling robustness but fail to capture interaction information among different channels due to the absence of inter-variable interactions. In contrast, models adopting the CM strategy [17], [25], [35] strengthen correlation modeling but tend to learn irrelevant random noise or spurious correlations during interaction due to neglecting

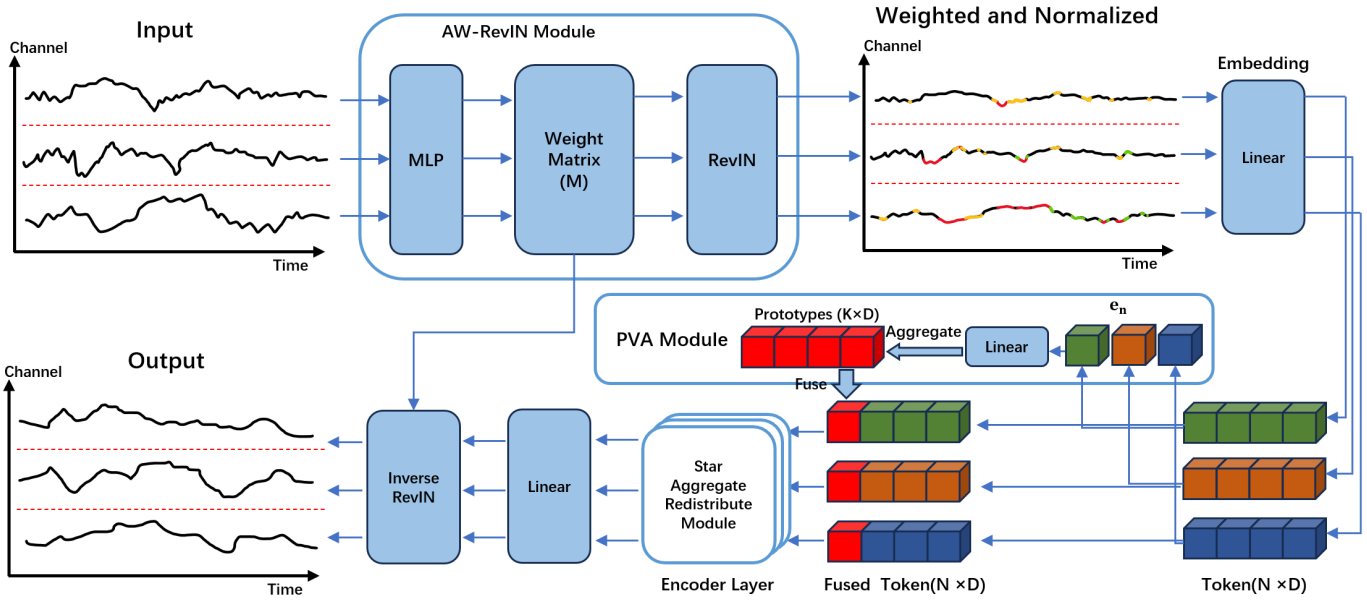

Fig. 2. The framework diagram of the AW-PVA model, where different colors in the "Weighted and Normalized" component represent varying weights assigned to different data sampling points.

behavioral pattern disparities among different variables [28]. Prototype learning, as a classical representation alignment mechanism, centers on learning a set of representative prototypical vectors to characterize complex data distributions. Research by [27] demonstrates that constructing multiple local prototypes combined with embedding adaptation mechanisms can effectively perform identity alignment and reconstruction of heterogeneous samples in the latent space.

Building upon these works, we design the AW-PVA model. Specifically, the AW-RevIN module addresses the distribution distortion caused by equal-weight contribution statistics through dynamically adjusting statistical weights of sampling points in conjunction with a sparsity penalty term. Furthermore, drawing on the principles of prototype learning, we design the PVA module to provide a set of standardized behavioral templates for diverse variables. This design avoids the blind coupling of variables with distinct evolutionary patterns, effectively filtering out spurious correlations during multivariate interactions and significantly enhancing the model's representation capacity.

III. STRUCTURE OVERVIEW

Problem Formulation. Given historical observation sequence $\mathbf{X} = \{x_1, \ldots, x_L\} \in \mathbb{R}^{L \times N}$, where L denotes the lookback window length and N denotes the number of variables (channels), with variables typically corresponding to observation indicators of different dimensions, the objective of MTSF is to establish a mapping model that predicts future numerical sequence $\mathbf{Y} = \{x_{L+1}, \ldots, x_{L+S}\} \in \mathbb{R}^{S \times N}$ for the next S time steps based on historical observations $\mathbf{X}$.

Model Overview. The overall architecture of the model is illustrated in Figure 2. First, the model employs the AW-RevIN module to learn the confidence weights of sample points in the input, producing weighted and normalized data along with a corresponding weight matrix $\mathbf{M}$. Subsequently, an inverted embedding approach is used to map the historical observations of each variable into independent tokens. Then, the PVA module injects physical semantics and evolutionary pattern information into each variable token. These tokens are then processed by the STAR encoder, which efficiently models global spatio-temporal interactions across variables via an aggregation-distribution mechanism. Finally, predicted values are generated through a linear projection layer, and the model's final output is obtained by performing de-normalization using the weight matrix $\mathbf{M}$ preserved during the normalization stage.

A. Adaptive Weighted RevIN(AW-RevIN)

Normalization is a prevalent technique for alleviating distribution shift in time series. Existing works [17]–[19], [24] rely on the "equal-weight contribution" assumption for normalization; however, this approach is sensitive to high noise or outliers, where local disturbances can distort distribution estimation and degrade feature representation. We posit that distribution contributions vary across different time points. Accordingly, we propose the AW-RevIN module built upon RevIN [21], which optimizes distribution estimation through dynamically adjusting sampling point weights.

The module first employs a lightweight MLP to learn a weight coefficient matrix $\mathbf{M} \in [0,1]^{B \times L \times N}$ that is strongly correlated with the input sequence. Given the input sequence $\mathbf{X} \in \mathbb{R}^{B \times L \times N}$, where B denotes the batch size, the process of assigning the weight coefficients $\mathbf{M}$ is formulated as follows:

$$M = \text{Sigmoid}\left(\text{MLP}\left(X^{\top}\right)\right)^{\top} \tag{1}$$

The weight matrix $\mathbf{M}$ serves as a reliability metric for each observation point, enabling automatic identification of repre-

sentative steady-state regions in the sequence. Furthermore, to prevent sampling points with near-zero weights from being misclassified as low-amplitude true observations during normalization, we incorporate $\mathbf{M}$ as a confidence factor into the statistical computation, constructing the weighted mean μ and weighted standard deviation σ:

$$\mu = \frac{\sum_{i=1}^{L} (X_i \odot M_i)}{\sum_{i=1}^{L} M_i + \epsilon}, \quad \sigma = \sqrt{\frac{\sum_{i=1}^{L} M_i \odot (X_i - \mu)^2}{\sum_{i=1}^{L} M_i + \epsilon} + \epsilon} \tag{2}$$

Finally, the normalized representation $\hat{\mathbf{X}}$ is obtained using the calibrated statistics, and the final input features $\mathbf{X}_{\text{in}}$ are derived by combining with the weight matrix:

$$\hat{X} = \left(\frac{X - \mu}{\sigma} \right) \odot \gamma + \beta, \quad X_{in} = \hat{X} \odot M \tag{3}$$

This design dynamically modulates the participation degree of each sampling point in the normalization computation, thereby suppressing non-stationary noise while maintaining distributional fidelity.

B. Series embedding

We adopt the inverted embedding strategy from iTransformer [17] to map variables as independent tokens. Specifically, this layer treats each variable's observation sequence across the entire lookback window L as a distinct entity. Through linear projection, the full-length sequence information of each variable is mapped into a high-dimensional hidden space, transforming into feature vectors $\mathbf{H}_{\text{enc}} \in \mathbb{R}^{B \times N \times D}$ with dimension D, where "variable" is equivalent to "token". This process can be formulated as:

$$\mathbf{H}_{enc} = \text{Embedding}(\mathbf{X}_{in}) \tag{4}$$

C. Prototype Vector Augmentation (PVA)

This constitutes our core innovation for addressing the blind coupling problem of variables. Under the inverted embedding paradigm, although variables are treated as independent tokens, raw value embeddings struggle to express their physical semantics or categorical attributes. To this end, we construct a matrix $\mathbf{P} \in \mathbb{R}^{K \times D}$ containing K globally shared prototypical vectors, where each prototype vector represents a typical variable evolution behavior, thereby constituting the model's global knowledge base. For any variable $n \in \{1, \ldots, N\}$ in a batch, the feature injection process proceeds as follows:

First, the model constructs a learnable parameter matrix $\mathbf{E} \in \mathbb{R}^{N \times Q}$, where Q denotes the query dimension. For the n-th variable, we extract the identity embedding $\mathbf{e}_n = \text{Extract}(\mathbf{E}, n)$ containing the variable's unique attributes as the query key. Subsequently, similarity scores between $\mathbf{e}_n$ and the K prototypes are computed through a linear layer, followed by Softmax to generate weights $w_{n,k}$:

$$W_{n,k} = \text{Softmax}(\text{Linear}(\mathbf{e}_n)) \tag{5}$$

Here $w_{n,k} \in [0, 1]$ denotes the similarity between variable n and the k-th prototypical vector, which defines the coordinate distribution of the variable within the global knowledge base. Through weights $w_{n,k}$, the model extracts relevant pattern information and aggregates it into the variable-specific identity signature vector $\mathbf{s}_n$:

$$\mathbf{S}_n = \sum_{k=1}^{K} W_{n,k} \cdot \mathbf{P}_k \tag{6}$$

Finally, the identity vector $\mathbf{S}_n$ is injected into the original temporal features $\mathbf{H}_{enc,n}$:

$$\mathbf{H}_{final,n} = \mathbf{H}_{enc,n} + \eta \cdot \text{Dropout}(\mathbf{S}_n) \tag{7}$$

This yields the final representation-enhanced feature matrix $\mathbf{H}_{final} \in \mathbb{R}^{B \times N \times D}$, where η is a scaling factor that controls the intensity of the feature injection.

D. Global Interaction via the STAR Module

We adopt the STAR (Spatial-Temporal Adaptive Representation) encoder proposed in SOFTS [18] to capture the complex cross-variable correlations in multivariate time series. This module achieves efficient global spatial modeling through an "aggregation-distribution" mechanism. The process can be formulated as follows:

$$\mathbf{H}_{out} = \text{Linear}(\text{Concat}(\mathbf{H}_{final}, \text{Aggregate}(\mathbf{H}_{final}))) \tag{8}$$

where $\mathbf{H}_{out} \in \mathbb{R}^{B \times N \times D}$. The Aggregate$(\cdot)$ function is designed to extract a global core representation shared by all variables.

E. Projection Output and Inverse Normalization

Following the practice of traditional state-of-the-art models [17], [18], we employ a simple yet efficient linear projection layer during forecasting to map the hidden dimension D to the prediction length S, yielding the forecast output $\mathbf{Y}_{\text{out}} \in \mathbb{R}^{B \times S \times N}$. Finally, through the inverse normalization operation, the predictions are restored to the original scale using the weighted statistics (μ, σ) preserved during the preprocessing stage:

$$\mathbf{Y}_{out} = \text{Projection}(\mathbf{H}_{out}), \quad \mathbf{Y} = \mathbf{Y}_{out} \odot \sigma + \mu \tag{9}$$

F. Loss Function and Sparsity Regularization

To ensure the model can adaptively identify key temporal characteristics while preventing the weight matrix generator from collapsing to the trivial solution of uniform weights, we introduce a sparsity regularization term into the training objective. The final loss function $\mathcal{L}_{\text{total}}$ comprises the primary forecasting loss and the regularization term: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{reg}}$. Here, $\mathcal{L}_{\text{forecast}}$ employs Mean Squared Error (MSE) as the objective for the prediction task, measuring the discrepancy between model outputs $\hat{\mathbf{Y}}$ and ground-truth observations $\mathbf{Y}$. $\mathcal{L}_{\text{reg}}$ represents the spatiotemporal average of

TABLE I

ALL RESULTS ARE OBTAINED WITH A FIXED LOOKBACK WINDOW $L = 96$ AND PREDICTION HORIZONS $S \in \{96, 192, 336, 720\}$. **RED**: BEST, BLUE: SECOND BEST.

Dataset	Metric	AW-PVA(Ours)		TQNet [19]		Soatten [36]		Fsatten [36]		DUET [37]		SOFTS [18]		iTrans [17]		MSGNet [35]		FreTS [33]	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
electricity	96	**0.132**	**0.228**	0.134	0.229	0.138	0.234	0.134	0.231	0.145	0.233	0.143	0.233	0.148	0.240	0.165	0.274	0.183	0.269
	192	**0.149**	**0.244**	0.154	0.247	0.156	0.248	0.152	0.247	0.163	0.248	0.158	0.248	0.162	0.253	0.185	0.292	0.187	0.276
	336	**0.163**	**0.262**	0.169	0.264	0.171	0.266	0.167	0.264	0.175	0.263	0.178	0.269	0.178	0.269	0.197	0.304	0.202	0.292
	720	**0.186**	**0.286**	0.201	0.294	0.199	0.288	0.195	0.287	0.204	0.291	0.218	0.305	0.225	0.317	0.231	0.332	0.237	0.325
	avg	**0.157**	**0.255**	0.164	0.258	0.166	0.259	0.162	0.257	0.171	0.258	0.174	0.263	0.178	0.269	0.194	0.300	0.202	0.290
ETTh1	96	0.374	0.397	**0.371**	**0.393**	0.383	0.401	0.384	0.401	0.377	**0.393**	0.381	0.399	0.386	0.405	0.390	0.411	0.399	0.412
	192	**0.426**	**0.425**	0.428	0.426	0.440	0.433	0.435	0.429	0.429	**0.425**	0.435	0.431	0.442	0.435	0.443	0.442	0.453	0.443
	336	**0.467**	**0.446**	0.476	**0.446**	0.474	0.449	0.481	0.453	0.471	0.447	0.480	0.452	0.487	0.458	0.482	0.469	0.503	0.475
	720	**0.480**	**0.470**	0.487	**0.470**	0.491	0.477	0.484	0.473	0.496	0.480	0.499	0.488	0.503	0.491	0.496	0.488	0.596	0.565
	avg	**0.436**	**0.434**	0.440	**0.434**	0.447	0.440	0.446	0.439	0.443	0.436	0.448	0.442	0.454	0.447	0.452	0.452	0.487	0.473
ETTh2	96	**0.293**	**0.343**	0.295	**0.343**	0.301	0.352	0.298	0.350	0.296	0.345	0.297	0.347	0.297	0.349	0.329	0.371	0.350	0.403
	192	**0.367**	0.393	**0.367**	0.393	0.384	0.402	0.381	0.401	0.368	**0.389**	0.373	0.394	0.380	0.400	0.402	0.414	0.472	0.475
	336	**0.405**	**0.425**	0.417	0.427	0.425	0.434	0.419	0.432	0.411	**0.425**	0.410	0.426	0.428	0.432	0.440	0.445	0.564	0.528
	720	0.414	**0.433**	0.433	0.446	0.429	0.447	0.426	0.445	0.414	0.434	**0.411**	**0.433**	0.427	0.445	0.480	0.477	0.815	0.654
	avg	**0.369**	**0.398**	0.378	0.402	0.384	0.408	0.381	0.407	0.372	**0.398**	0.372	0.400	0.383	0.406	0.412	0.426	0.550	0.515
ETTm1	96	0.319	0.357	**0.311**	**0.353**	0.329	0.366	0.330	0.370	0.324	0.357	0.325	0.361	0.334	0.368	0.319	0.366	0.339	0.374
	192	0.365	0.382	**0.356**	**0.378**	0.371	0.387	0.373	0.392	0.369	0.379	0.375	0.389	0.377	0.391	0.377	0.397	0.382	0.397
	336	0.398	0.408	**0.390**	**0.401**	0.402	0.408	0.404	0.411	0.404	0.411	0.404	**0.401**	0.405	0.412	0.417	0.422	0.421	0.426
	720	0.456	0.447	**0.452**	0.440	0.474	0.447	0.469	0.447	0.463	**0.437**	0.466	0.447	0.491	0.459	0.487	0.463	0.485	0.462
	avg	0.384	0.398	**0.377**	**0.393**	0.394	0.402	0.394	0.405	0.390	**0.393**	0.392	0.402	0.407	0.409	0.400	0.412	0.406	0.414
ETTm2	96	**0.172**	**0.253**	0.173	0.256	0.179	0.264	0.180	0.265	0.174	0.255	0.180	0.261	0.180	0.264	0.182	0.266	0.190	0.282
	192	**0.238**	**0.298**	**0.238**	**0.298**	0.246	0.306	0.245	0.306	0.243	0.302	0.246	0.306	0.250	0.309	0.248	0.306	0.260	0.329
	336	**0.298**	**0.337**	0.301	0.340	0.312	0.349	0.307	0.347	0.304	0.341	0.319	0.352	0.311	0.348	0.312	0.346	0.373	0.405
	720	0.401	0.398	**0.397**	**0.396**	0.411	0.405	0.412	0.406	0.402	0.399	0.405	0.401	0.412	0.407	0.414	0.404	0.517	0.499
	avg	**0.277**	**0.321**	**0.277**	0.322	0.287	0.331	0.286	0.331	0.280	0.324	0.287	0.330	0.288	0.332	0.289	0.330	0.335	0.378
solar	96	0.175	**0.229**	**0.173**	0.233	0.197	0.238	0.196	0.232	0.200	0.231	0.200	0.230	0.203	0.237	0.210	0.246	0.192	0.225
	192	0.204	**0.252**	**0.199**	0.257	0.235	0.267	0.227	0.256	0.228	0.252	0.229	0.253	0.233	0.261	0.265	0.290	0.229	0.252
	336	0.216	**0.262**	**0.211**	0.263	0.251	0.278	0.247	0.272	0.262	0.273	0.243	0.269	0.248	0.273	0.294	0.318	0.242	0.269
	720	0.219	**0.267**	**0.209**	0.270	0.253	0.281	0.250	0.276	0.258	0.285	0.245	0.272	0.249	0.275	0.285	0.315	0.240	0.272
	avg	0.203	**0.252**	**0.198**	0.255	0.234	0.266	0.230	0.259	0.237	0.260	0.229	0.256	0.233	0.261	0.263	0.292	0.225	0.254
weather	96	**0.151**	**0.198**	0.157	0.200	0.161	0.205	0.166	0.208	0.163	0.202	0.166	0.208	0.174	0.214	0.163	0.212	0.184	0.239
	192	**0.198**	**0.244**	0.206	0.245	0.209	0.249	0.212	0.251	0.218	0.252	0.217	0.253	0.221	0.254	0.211	0.254	0.223	0.275
	336	**0.254**	**0.285**	0.262	0.287	0.264	0.292	0.268	0.293	0.274	0.294	0.282	0.300	0.278	0.296	0.273	0.299	0.272	0.316
	720	**0.333**	**0.341**	0.344	0.342	0.346	0.346	0.350	0.348	0.349	0.343	0.356	0.351	0.358	0.347	0.351	0.348	0.340	0.363
	avg	**0.234**	**0.267**	0.242	0.268	0.245	0.273	0.249	0.275	0.251	0.272	0.255	0.278	0.257	0.277	0.249	0.278	0.254	0.298

the weight matrix $\mathbf{M}$, $\text{Avg}(\mathbf{M})$ (a form of L_1 norm), with its complete formulation as follows:

$$\mathcal{L}_{\text{total}} = \frac{1}{S \times N} \sum_{t=1}^{S} \sum_{n=1}^{N} (\hat{y}_{t,n} - y_{t,n})^2 + \lambda \frac{1}{L \times N} \sum_{i=1}^{L} \sum_{j=1}^{N} m_{i,j} \tag{10}$$

where $m_{i,j} \in [0, 1]$. By minimizing $\text{Avg}(\mathbf{M})$, the model is guided to precisely capture core sampling points most representative of the underlying distributional patterns. In our experiments, the penalty coefficient λ is set to 0.01 to balance forecasting accuracy with the sparsity of regularization.

IV. EXPERIMENTS

Datasets. To comprehensively evaluate the performance of the proposed AW-PVA, we conducted extensive experiments on seven widely-used real-world datasets, including ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Weather, and Solar-Energy.

A. Forecasting Results

Baselines. To evaluate the performance of AW-PVA, we compare it against several representative models recently proposed in the field, including TQNet [19], Soatten [36], Fsatten [36], DUET [37], iTransformer [17], SOFTS [18], FreTS [33], and MSGNet [35].

Experimental Setup. Long-term forecasting tasks follow the benchmark configurations of Informer [12] and SCINet [30]: lookback window $L = 96$, prediction horizons $S \in \{96, 192, 336, 720\}$. Performance is evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE) (**lower is better**). Main results of baselines are sourced from TQNet [19] and iTransformer [17].

Implementation Details. Our primary hyperparameters follow the configurations in iTransformer [17]. The initial weight bias is set to $b_{\text{init}} \in \{0.0, 1.0\}$ (corresponding to initial activation values of approximately 50% and 73%). The number of prototype vectors K is determined by the number of variables N: for the ETT datasets with fewer variables,

$K \in \{8, 16, 32\}$, while for datasets with more variables, $K \in \{32, 64, 128\}$.

Main Results. As shown in Table I, AW-PVA achieves optimal or suboptimal forecasting performance across all 7 datasets. Furthermore, compared to previous state-of-the-art models, AW-PVA demonstrates significant improvements. Specifically, on the Weather dataset, the average MSE decreases from 0.242 to 0.234, representing a relative reduction of approximately 3.3%. On the Electricity dataset, the average MSE drops from 0.164 to 0.157, achieving a relative reduction of 4.2%. These improvements fully validate the superior performance and broad applicability of AW-PVA in MTSF tasks.

B. Ablation Study

Main Component Ablation Study. To evaluate the contributions of individual AW-PVA components, we present ablation experiment results in Table II(a).We decompose the model into the following variants: (1) w/o-AW-RevIN, employing standard normalization akin to iTransformer as a replacement; (2) w/o-PVA, removing the prototypical vector augmentation module; and (3) w/o AW-PVA, representing complete ablation approximating SOFTS [18]. Comparative analysis demonstrates that the complete model achieves significant improvements over ablation baselines, with average MSE relatively reduced by 8.2% to 13.2%. Furthermore, to validate the effectiveness of adaptive weighting, we additionally introduce a control group employing vanilla RevIN [21] (denoted as with-RevIN). Comparison with w/o-PVA demonstrates that the adaptive weighting mechanism extracts stationary features more effectively, significantly outperforming the traditional RevIN architecture, particularly on Solar and Weather datasets.

TABLE II
MAIN RESULTS OF ABLATION EXPERIMENTS. ALL RESULTS ARE OBTAINED WITH A FIXED LOOKBACK WINDOW $L = 96$, AVERAGED ACROSS PREDICTION HORIZONS $S \in \{96, 192, 336, 720\}$. **RED**: BEST, BLUE: SECOND BEST.

Group	Model	electricity		solar		weather	
		MSE	MAE	MSE	MAE	MSE	MAE
a	AW-PVA	**0.157**	**0.255**	**0.203**	**0.252**	**0.234**	**0.268**
	w/o AW-RevIN	0.164	0.257	0.229	0.260	0.241	0.269
	w/o PVA	0.162	0.258	0.217	0.255	0.236	0.272
	w/o AW-PVA	0.176	0.264	0.234	0.262	0.255	0.277
	with-RevIN [21]	0.167	0.258	0.229	0.265	0.246	0.273
	SOFTS [18]	0.174	0.263	0.229	0.256	0.255	0.278
b	AW-PVA-iTrans	**0.160**	**0.259**	**0.203**	**0.252**	**0.238**	**0.271**
	iTransformer [17]	0.178	0.269	0.233	0.261	0.257	0.277

Component Transferability Analysis. To validate the universality of AW-PVA core components, we integrate them into iTransformer [17]. Experimental results are presented in Table II(b). Comparative analysis demonstrates that introducing AW-PVA components significantly enhances forecasting accuracy, with average MSE reduced by 7.3%–12.8%. This fully demonstrates that AW-PVA core components possess excellent compatibility.

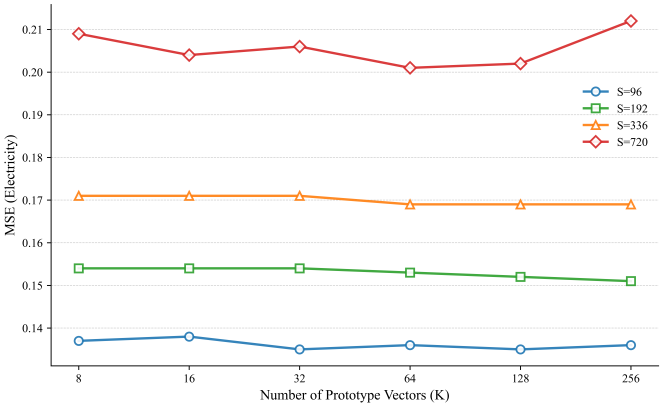

Fig. 3. Sensitivity of Hyperparameter K

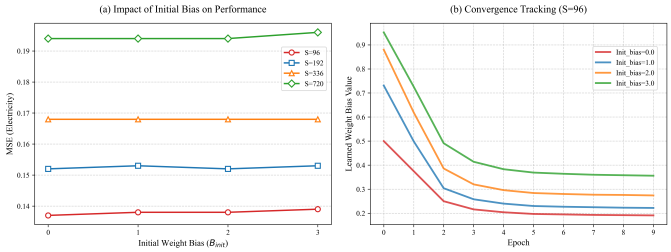

Fig. 4. Analysis of Hyperparameter Initial Weight Bias.

Hyperparameter K Analysis. Figure 3 illustrates the performance on the Electricity dataset with $K \in \{8, 16, 32, 64, 128, 256\}$ across prediction horizons $S \in \{96, 192, 336, 720\}$. The results reveal that the value of K only slightly affects model accuracy when the prediction horizon is 720, with minimal impact in other cases. Furthermore, optimal performance is achieved when K = 128. This observation aligns with our expectation that for scenarios with large variable dimension N, the empirical optimum for K typically resides within the range $[N/4, N/3]$.

Analysis of Hyperparameter Initial Bias. We evaluate the sensitivity on the Electricity dataset with $b_{\text{init}} \in \{0.0, 1.0, 2.0, 3.0\}$ (corresponding to activation values $\{0.500, 0.731, 0.880, 0.952\}$). As shown in Figure 4 (left), across prediction horizons $S \in \{96, 192, 336, 720\}$, significantly different initial weights ultimately converge to comparable prediction accuracy with minimal error variance. Furthermore, as illustrated in Figure 4 (right), when $S = 96$, although converged bias values do not collapse to a single point, they all guide the model to achieve equivalent performance. This indicates that the module possesses strong adaptability and initialization robustness, capable of autonomously optimizing weight distributions without relying on specific initial values.

V. CONCLUSION

This paper proposes the AW-PVA model, which enhances distributional fidelity by dynamically adjusting statistical weights of sampling points to filter extreme noise, while

employing K globally shared prototypical vectors to resolve blind coupling among heterogeneous variables, thereby augmenting the model's representational capability. Experimental results demonstrate that AW-PVA outperforms existing state-of-the-art baselines, with ablation studies and hyperparameter analyses further validating the universality of the proposed modules and their effectiveness as plug-and-play components. In summary, this study presents an innovative solution for time series forecasting that achieves both distributional stability and feature discriminability, offering significant academic reference value.

REFERENCES

[1] Y. Peng, M. Lei, J. Guo, and X. Peng, "Multiresolution analysis and forecasting of mobile communication traffic," *Chinese journal of electronics*, vol. 22, no. 2, pp. 373–376, 2013.

[2] Y. Wang, H. Wu, J. Dong, Y. Liu, C. Wang, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," *arXiv preprint arXiv:2407.13278*, 2024.

[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[4] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, "Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective," *Advances in neural information processing systems*, vol. 36, pp. 69 638–69 660, 2023.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] A. Gruca, F. Serva, L. Lliso, P. Rípodas, X. Calbet, P. Herruzo, J. Pihrt, R. Raevskyi, P. Šimánek, M. Choma *et al.*, "Weather4cast at neurips 2022: Super-resolution rain movie prediction under spatio-temporal shifts," in *NeurIPS 2022 competition track*. PMLR, 2023, pp. 292–313.

[7] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," *Advances in neural information processing systems*, vol. 31, 2018.

[8] A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[9] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 95–104.

[10] S. Bai, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[11] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, vol. 379, no. 2194, 2021.

[12] 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.

[13] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

[14] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[15] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International conference on learning representations*, 2021.

[16] 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.

[17] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[18] L. Han, X.-Y. Chen, H.-J. Ye, and D.-C. Zhan, "Softs: Efficient multivariate time series forecasting with series-core fusion," *Advances in Neural Information Processing Systems*, vol. 37, pp. 64 145–64 175, 2024.

[19] S. Lin, H. Chen, H. Wu, C. Qiu, and W. Lin, "Temporal query network for efficient multivariate time series forecasting," *arXiv preprint arXiv:2505.12917*, 2025.

[20] 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.

[21] 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.

[22] Z. Xu, A. Zeng, and Q. Xu, "Fits: Modeling time series with $10k$ parameters," *arXiv preprint arXiv:2307.03756*, 2023.

[23] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[24] Y. Nie, "A time series is worth 64words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[25] 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.

[26] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, "Segrnn: Segment recurrent neural network for long-term time series forecasting," *IEEE Internet of Things Journal*, 2025.

[27] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8808–8817.

[28] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "Timemixer: Decomposable multiscale mixing for time series forecasting," *arXiv preprint arXiv:2405.14616*, 2024.

[29] R. Ilbert, A. Odonnat, V. Feofanov, A. Virmaux, G. Paolo, T. Palpanas, and I. Redko, "Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention," *CoRR*, 2024.

[30] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "Scinet: Time series modeling and forecasting with sample convolution and interaction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.

[31] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, and T. Pfister, "Tsmixer: An all-mlp architecture for time series forecasting," *arXiv preprint arXiv:2303.06053*, 2023.

[32] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[33] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, "Frequency-domain mlps are more effective learners in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 36, pp. 76 656–76 679, 2023.

[34] L. Han, H.-J. Ye, and D.-C. Zhan, "The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 7129–7142, 2024.

[35] W. Cai, Y. Liang, X. Liu, J. Feng, and Y. Wu, "Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 10, 2024, pp. 11 141–11 149.

[36] H. Wu, "Revisiting attention for multivariate time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 20, 2025, pp. 21 528–21 535.

[37] X. Qiu, X. Wu, Y. Lin, C. Guo, J. Hu, and B. Yang, "Duet: Dual clustering enhanced multivariate time series forecasting," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2025, pp. 1185–1196.