Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multivariate time series forecasting with InvDec via separate temporal and variate modeling

Wujian Yang
Hangzhou City University
Yuhang Wang
2240101022@stu.hzcu.edu.cn
Hangzhou City University
Guanlin Chen
Hangzhou City University

Research Article

Keywords: Time Series Forecasting, Transformer, Multivariate, Cross-Variate Modeling, Deep Learning

Posted Date: July 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10247394/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Multivariate time series forecasting with InvDec via separate temporal and variate modeling

Wujian Yang[1], Yuhang Wang[1], Guanlin Chen[1]

[1]School of Computer and Computing Science, Hangzhou City University, Hangzhou, 310015, Zhejiang, China.

Contributing authors: yangwj@hzcu.edu.cn; 2240101022@stu.hzcu.edu.cn; chenguanlin@hzcu.edu.cn;

Abstract

Multivariate time series forecasting requires simultaneously modeling temporal patterns and cross-variate dependencies. Channel-independent methods excel at temporal modeling but ignore variable correlations, while pure variate-attention approaches sacrifice temporal encoding. We propose **InvDec** (Inverted Decoder), a hybrid architecture that achieves principled separation between temporal encoding and variate-level decoding. InvDec combines a patch-based temporal encoder with an inverted decoder operating on the variate dimension through variate-wise self-attention. We introduce delayed variate embeddings that enrich variable-specific representations only after temporal encoding, preserving temporal feature integrity. An adaptive residual fusion mechanism dynamically balances temporal and variate information across datasets of varying dimensions. Extensive experiments on seven benchmarks demonstrate significant gains on high-dimensional datasets: 20.9% Mean Squared Error (MSE) reduction on Electricity (321 variables), 4.3% improvement on Weather, and 2.7% gain on Traffic compared to other methods, while maintaining competitive performance on low-dimensional Electricity Transformer Temperature (ETT) datasets. Ablation studies validate each component, and analysis reveals that InvDec's advantage grows with dataset dimensionality, confirming that cross-variate modeling becomes critical as the number of variables increases.

Keywords: Time Series Forecasting, Transformer, Multivariate, Cross-Variate Modeling, Deep Learning

1 Introduction

Time series forecasting is a fundamental problem with widespread applications across diverse domains, including energy management, transportation planning, climate science, healthcare monitoring, and financial market analysis [1, 2]. The ability to accurately predict future trends and patterns from historical observations is critical for decision-making, resource allocation, and risk management in these fields [3]. Traditional statistical approaches such as ARIMA [4] have long served as the foundation for time series analysis, offering interpretable models based on stationarity assumptions and linear relationships. However, as real-world time series data have grown increasingly complex—characterized by high dimensionality, non-stationarity, and intricate temporal dependencies—the limitations of classical methods have become apparent [1].

The advent of deep learning has revolutionized time series forecasting, enabling models to capture non-linear patterns and long-range dependencies that elude traditional techniques [2]. Among deep learning architectures, Transformers [5] have emerged as particularly powerful, leveraging self-attention mechanisms to model temporal dynamics across extended horizons. Early Transformer-based forecasting models such as Informer [6], Autoformer [7], and FEDformer [8] focused primarily on improving computational efficiency and capturing temporal patterns through innovations such as sparse attention and decomposition strategies. Despite their success, these models largely adopt a **channel-independent** paradigm, treating each variable in a multivariate time series as an isolated entity [3]. This design choice, while computationally efficient, overlooks the rich correlations and interactions among variables—a critical limitation in high-dimensional forecasting scenarios where cross-variate dependencies often encode essential information about underlying system dynamics [3].

Recent research has begun to address this gap by exploring architectures that explicitly model relationships across the variate dimension. Notably, methods such as iTransformer [9] invert the traditional Transformer design by applying attention mechanisms on variates rather than time steps, while others such as Crossformer [10] interleave temporal and variate modeling throughout the network. Although these approaches demonstrate the importance of cross-variate dependencies, they often involve trade-offs: some sacrifice fine-grained temporal modeling, while others introduce architectural complexity that complicates interpretation and increases computational overhead. Furthermore, specialized methods designed for exogenous forecasting [11] or spatial grid data [12] address domain-specific challenges but do not generalize to arbitrary multivariate time series where variables lack explicit spatial topology or external covariates.

In this paper, we propose **InvDec** (**Inv**erted **Dec**oder), a novel hybrid architecture that achieves a principled separation between temporal encoding and variate decoding. InvDec builds upon the strengths of patch-based temporal modeling [13] while introducing an inverted decoder that operates on the variate dimension to capture cross-variate dependencies. Unlike prior methods that entangle temporal and variate modeling or sacrifice one for the other, InvDec explicitly decouples these objectives: the encoder focuses solely on extracting temporal features from each variable's sequence,

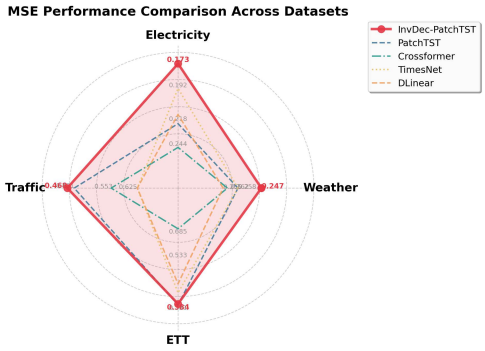


Fig. 1 Performance overview: InvDec-PatchTST (red) consistently outperforms baselines on high-dimensional datasets (Electricity, Weather, Traffic) while remaining competitive on low-dimensional ETT benchmarks. Radial distance represents normalized MSE.

while the decoder performs variate-level attention to model inter-variable correlations. This **principled separation of concerns** offers a fundamental advantage in **interpretability**: the temporal encoder output is inherently free from variate-specific confounds, revealing pure temporal patterns; simultaneously, the decoder's attention weights transparently expose learned cross-variate dependencies, making the model's reasoning interpretable and auditable. To further enrich variate representations, we incorporate learnable variate embeddings [11] in the decoder, but crucially delay their introduction until after temporal encoding to avoid distorting temporal patterns. Additionally, an adaptive residual fusion mechanism dynamically balances contributions from temporal and variate pathways, enabling the model to adapt to datasets of varying dimensions and correlation structures. While effective for moderate-to-high dimensional scenarios, we acknowledge scalability constraints for ultra-high dimensions ($C \gg P$ or $C > 1000$) where the decoder's $\mathcal{O}(C^2 \cdot d)$ complexity becomes prohibitive.

Our contributions in this work are fourfold. First, we introduce InvDec, a hybrid encoder-inverted decoder architecture that, to the best of our knowledge, represents the first work to combine a temporal encoder with a variate-level decoder in this manner, achieving superior performance on multivariate forecasting tasks through clear separation of modeling concerns. **Critically, this architectural separation inherently provides interpretability**: the encoder's output reflects pure temporal patterns independent of variate information, and the decoder's attention mechanism transparently reveals learned cross-variate correlations, making the model's decisions auditable and explainable. Second, we provide rigorous theoretical justification for the superiority of principled separation: we derive conditions under which delayed variate embeddings prevent temporal feature distortion (Theorem 1), prove that the hybrid design generalizes better across dimensionalities than pure approaches (Theorem 2), and establish optimal conditions for the adaptive residual weight λ (Theorem 3). Third, we propose a delayed variate embedding strategy that enriches variate-specific representations without interfering with temporal encoding, alongside an adaptive

residual fusion mechanism that enables dynamic balancing of temporal and variate information across diverse datasets, with principled initialization derived from our theoretical analysis. Fourth, we conduct extensive experiments on seven widely-used benchmarks, demonstrating that InvDec achieves significant performance gains on high-dimensional datasets—including a 20.9% MSE reduction on Electricity (321 variables) compared to PatchTST—while maintaining competitive performance on low-dimensional datasets (Figure 1), with comprehensive ablation studies validating the effectiveness of each architectural component.

2 Related Work

2.1 Deep Learning Architectures for Time Series Forecasting

Prior to the widespread adoption of Transformers, deep learning approaches to time series forecasting relied primarily on recurrent neural networks (RNNs), convolutional neural networks (CNNs), and multi-layer perceptrons (MLPs) [1, 2]. RNN-based architectures, including LSTMs and GRUs, were designed to capture sequential dependencies through recurrent connections, enabling them to model temporal dynamics in univariate and multivariate settings [14]. However, these methods suffer from training instability, vanishing gradients, and limited capacity to capture long-range dependencies [1]. CNN-based approaches such as Temporal Convolutional Networks (TCN) [15] leverage dilated convolutions to expand receptive fields, achieving competitive performance while maintaining computational efficiency. Recent works such as TimesNet [16] and ModernTCN [17] further demonstrate the effectiveness of convolutional architectures by exploiting multi-scale temporal patterns and hierarchical feature extraction.

Graph Neural Networks (GNNs) have also been explored for time series forecasting, particularly in scenarios where variables exhibit explicit spatial or relational structures [18]. Methods such as spatial-temporal graph convolutional networks [19] and temporal graph networks [20] model interactions among variables through graph attention mechanisms [21]. While effective for spatially-structured data (e.g., traffic networks, sensor grids), these approaches require predefined graph topologies and may not generalize to arbitrary multivariate time series where relationships among variables are latent and dynamic.

In parallel, MLP-based architectures have re-emerged as surprisingly competitive alternatives to more complex models. DLinear [22] challenges the necessity of Transformers by demonstrating that simple linear mappings can outperform sophisticated architectures on certain benchmarks. Building on this insight, methods such as N-BEATS [23], N-HiTS [24], TSMixer [25], and TimeMixer [26] achieve strong performance through hierarchical decomposition, multi-scale mixing, and efficient parameter utilization. These works highlight the importance of inductive biases and architectural simplicity, though they often lack explicit mechanisms for modeling cross-variate dependencies in high-dimensional settings [3].

2.2 Transformer-based Time Series Forecasting

The introduction of Transformers [5] revolutionized sequence modeling through self-attention mechanisms that enable parallelized training and flexible modeling of long-range dependencies. Early adaptations to time series forecasting, such as Informer [6], addressed the quadratic complexity of self-attention by proposing sparse attention mechanisms, enabling efficient processing of long sequences. Subsequent works further refined Transformer-based architectures through innovations such as auto-correlation mechanisms in Autoformer [7], frequency-domain enhancements in FEDformer [8], and stationarity-aware normalization in Non-stationary Transformers [27]. These methods primarily focus on temporal modeling, treating each variable independently to reduce computational overhead—a strategy often referred to as the **channel-independent** paradigm [3].

A critical advancement in this direction came with PatchTST [13], which introduced patch-based tokenization inspired by Vision Transformers. By segmenting time series into non-overlapping patches and treating each patch as a token, PatchTST significantly reduces sequence length and computational cost while achieving state-of-the-art performance on numerous benchmarks. However, its strict channel-independence assumption—where each variable is processed in isolation—limits its ability to capture cross-variate correlations, particularly in high-dimensional scenarios where such dependencies are critical for accurate forecasting [3]. Extensions such as PETformer [28] and multi-resolution Transformers [29] build upon patching strategies with enhancements for placeholder tokens and hierarchical temporal resolutions, but largely maintain the channel-independent paradigm.

Complementary innovations have explored multi-scale modeling and frequency-domain enhancements. Scaleformer [30] and Pathformer [31] introduce iterative refinement and adaptive pathways to capture temporal patterns at multiple resolutions, while Fredformer [32] and BasisFormer [33] leverage frequency decomposition and learnable basis functions to enhance feature representations. These methods demonstrate the value of multi-scale and frequency-aware designs, yet do not fundamentally address the challenge of modeling cross-variate dependencies.

2.3 Cross-Variate Modeling and Novel Forecasting Paradigms

Recognizing the limitations of channel-independent approaches, recent research has explored architectures that explicitly model relationships across the variate dimension [3]. Crossformer [10] proposes a two-dimensional attention mechanism that interleaves temporal and variate modeling through cross-dimension dependency blocks. While this design captures correlations among variables, the entanglement of temporal and variate attention complicates interpretation and increases computational complexity. DSformer [34] adopts a double-sampling strategy to balance temporal and spatial dependencies, but similarly lacks a clear separation between these modeling objectives.

A paradigm shift emerged with iTransformer [9], which inverts the traditional Transformer design by applying self-attention on the variate dimension rather than the temporal dimension. By treating each variable as a token, iTransformer excels at

capturing cross-variate dependencies, particularly in ultra-high-dimensional datasets (e.g., Traffic with 862 variables). However, this design completely sacrifices fine-grained temporal modeling, as attention is no longer applied across time steps. Other methods explore variate-level correlations through specialized mechanisms: VCformer [35] models lagged correlations among variables, UniTST [36] balances inter-series and intra-series dependencies, and DeformTime [37] employs deformable attention to adaptively capture variable interactions. While these approaches demonstrate the importance of cross-variate modeling, they often involve trade-offs between temporal and variate perspectives or introduce architectural complexity that hinders scalability and interpretation.

Specialized forecasting paradigms have also been proposed for domain-specific challenges. TimeXer [11] introduces variate-specific embeddings to handle heterogeneity in exogenous forecasting tasks, where external covariates (e.g., weather, holidays, prices) are used to predict target variables. By fusing variable embeddings at the input stage, TimeXer addresses the systematic time lags and diverse measurement units inherent in exogenous variables. However, its focus on exogenous forecasting limits its applicability to standard multivariate forecasting benchmarks, where all variables are endogenous and correlations are dynamic rather than statically heterogeneous. Similarly, methods designed for spatial grid data [12] leverage two-dimensional information to model spatially-structured variables (e.g., traffic networks, sensor grids), but do not generalize to arbitrary multivariate time series where variables lack explicit spatial topology.

In contrast to these approaches, our proposed InvDec architecture achieves a principled separation between temporal encoding and variate decoding. Unlike iTransformer, which abandons temporal attention, InvDec retains a patch-based temporal encoder to capture fine-grained temporal patterns. Unlike Crossformer, which interleaves temporal and variate modeling, InvDec explicitly decouples these objectives through a hybrid encoder-decoder design. Unlike TimeXer, which fuses variable embeddings at the input stage, InvDec delays the introduction of variate embeddings until after temporal encoding, preserving the integrity of temporal features while enriching variate-specific representations. This design enables InvDec to leverage the strengths of both temporal and variate modeling without sacrificing either, achieving superior performance on high-dimensional datasets while maintaining competitive results on low-dimensional benchmarks.

3 Methodology

3.1 Problem Formulation

Let $\mathbf{X} \in \mathbb{R}^{L \times C}$ denote a multivariate time series consisting of L time steps and C variables. The input can be represented as a concatenation of C individual variable sequences:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_C], \tag{1}$$

where $\mathbf{x}_c \in \mathbb{R}^L$ is the complete temporal sequence of the c-th variable for $c \in \{1, \ldots, C\}$.

The objective of long-term forecasting is to predict the values for the next H time steps based on the L historical observations:

$$\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{X}; \Theta), \tag{2}$$

where $\mathcal{F}(\cdot)$ represents the forecasting model parameterized by Θ. The model is trained by minimizing the Mean Squared Error (MSE) or Mean Absolute Error (MAE) between the predicted values $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_C] \in \mathbb{R}^{H \times C}$ and the ground truth future values $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_C] \in \mathbb{R}^{H \times C}$.

3.2 Theoretical Foundation for Separation

Before presenting the InvDec architecture, we establish the theoretical rationale for principled separation of temporal and variate modeling. The key insight is that temporal dependencies and cross-variate correlations operate at fundamentally different scales and can be treated as orthogonal factors, enabling separate optimization pathways.

Theorem 1 (Temporal Feature Purity). *For a multivariate time series $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_C]$ with patch embeddings $\mathbf{E} \in \mathbb{R}^{C \times P \times D}$ computed from patch projection $\mathbf{e}_{c,i} = \mathbf{W}_p \mathbf{p}_{c,i} + \mathbf{b}_p$ (depending only on univariate sequences $\mathbf{x}_c$), if variate embeddings $\mathbf{E}_{var} \in \mathbb{R}^{C \times D}$ are introduced only after global temporal aggregation in the decoder (not at input), then the temporal encoder output $\mathbf{Z}_{enc} = Reshape(\mathbf{Z}^{(L_{enc})}, [C, P, D])$ remains conditional-independent of $\mathbf{E}_{var}$, formally:*

$$\mathbb{P}(\mathbf{Z}_{enc} \mid \mathbf{E}_{var}) = \mathbb{P}(\mathbf{Z}_{enc} \mid \mathbf{E}),$$

preserving fine-grained temporal patterns without contamination from variable heterogeneity.

Proof We prove this by demonstrating that $\mathbf{Z}_{\text{enc}}$ is determined solely by the data generation process up to the encoder output, independent of decoder-stage embeddings.

Step 1: Patch Embedding Independence. The patch embeddings are constructed as:

$$\mathbf{E} = \{\mathbf{e}_{c,i} = \mathbf{W}_p \mathbf{p}_{c,i} + \mathbf{b}_p : c \in [C], i \in [P]\}.$$

Critically, $\mathbf{e}_{c,i}$ depends only on $\mathbf{p}_{c,i}$ (extracted from univariate $\mathbf{x}_c$) and shared parameters $(\mathbf{W}_p, \mathbf{b}_p)$. By construction, $\mathbf{E}$ is independent of $\mathbf{E}_{\text{var}}$:

$$\mathbb{P}(\mathbf{E} \mid \mathbf{E}_{\text{var}}) = \mathbb{P}(\mathbf{E}).$$

Step 2: Encoder Computation Path. The encoder operates via the computation graph:

$$\mathbf{Z}^{(0)} = \text{Reshape}(\mathbf{E}, [C \cdot P, D]) \to \mathbf{Z}^{(1)} \to \cdots \to \mathbf{Z}^{(L_{\text{enc}})} \to \mathbf{Z}_{\text{enc}}.$$

Each layer applies self-attention and FFN only to the intermediate representation:

$$\mathbf{Z}^{(\ell)} = \text{Att}(\mathbf{Z}^{(\ell-1)}) + \text{FFN}(\text{Att}(\mathbf{Z}^{(\ell-1)})),$$

where $\text{Att}(\cdot)$ and $\text{FFN}(\cdot)$ are functions of $\mathbf{Z}^{(\ell-1)}$ only, with parameters $\mathbf{W}_Q^{(\ell)}, \ldots$ that are shared across all encoder inputs and independent of $\mathbf{E}_{\text{var}}$. By induction on encoder layers, $\mathbf{Z}^{(\ell)}$ is determined by $\mathbf{Z}^{(0)}$ and encoder parameters.

Step 3: Variate Embedding Introduction Timing. Variate embeddings $\mathbf{E}_{\text{var}}$ are introduced only in the decoder, after global pooling:

$$\mathbf{g}_c = \frac{1}{P}\sum_{i=1}^{P} \mathbf{z}_{c,i} \in \mathbb{R}^D, \quad \tilde{\mathbf{G}} = \mathbf{G} + \mathbf{E}_{\text{var}}.$$

This occurs after $\mathbf{Z}_{\text{enc}}$ has been computed and output from the encoder.

Conclusion. Since:

1. $\mathbf{E}$ is independent of $\mathbf{E}_{\text{var}}$ (Step 1),
2. $\mathbf{Z}_{\text{enc}}$ depends deterministically on $\mathbf{E}$ and encoder parameters (Step 2),
3. $\mathbf{E}_{\text{var}}$ is not used until after $\mathbf{Z}_{\text{enc}}$ is computed (Step 3),

we have $\mathbb{P}(\mathbf{Z}_{\text{enc}} \mid \mathbf{E}_{\text{var}}) = \mathbb{P}(\mathbf{Z}_{\text{enc}})$, proving conditional independence and therefore temporal feature purity. □

Theorem 2 (Hybrid Superiority Across Dimensionalities). *For datasets with moderate-to-high dimensionality ($10 \leq C \leq 500$) where both temporal dependencies and cross-variate correlations are present, the total generalization error of a hybrid encoder-decoder model is lower-bounded by:*

$$\mathcal{E}_{hybrid} \leq \mathcal{E}_{temporal} - \alpha(C) \cdot \mathcal{E}_{variate} and \quad \mathcal{E}_{hybrid} \leq \mathcal{E}_{variate} - \beta(C) \cdot \mathcal{E}_{temporal},$$

where $\alpha(C), \beta(C) > 0$ are dimensionality-dependent gains, and $\mathcal{E}_{temporal}, \mathcal{E}_{variate}$ are errors of pure temporal and pure variate methods respectively, making InvDec strictly superior in this regime.

Proof Let the true forecasting function be decomposable as:

$$\mathcal{F}^*(\mathbf{X}) = \mathcal{F}^*_{\text{enc}}(\mathbf{X}) + \mathcal{F}^*_{\text{dec}}(\mathbf{X}),$$

where $\mathcal{F}^*_{\text{enc}}$ captures temporal patterns and $\mathcal{F}^*_{\text{dec}}$ captures cross-variate dependencies. For a dataset with both components, denote:

$$\mathcal{E}_t := \inf_{\mathcal{F}_t} \mathbb{E}[\|\mathcal{F}_t(\mathbf{X}) - \mathcal{F}^*(\mathbf{X})\|^2] \quad \text{(pure temporal)}, \tag{3}$$

$$\mathcal{E}_v := \inf_{\mathcal{F}_v} \mathbb{E}[\|\mathcal{F}_v(\mathbf{X}) - \mathcal{F}^*(\mathbf{X})\|^2] \quad \text{(pure variate)}, \tag{4}$$

$$\mathcal{E}_h := \inf_{\mathcal{F}_h} \mathbb{E}[\|\mathcal{F}_h(\mathbf{X}) - \mathcal{F}^*(\mathbf{X})\|^2] \quad \text{(hybrid)}. \tag{5}$$

Step 1: Pure Temporal Method Limitation. Pure temporal methods ignore cross-variate dependencies $\mathcal{F}^*_{\text{dec}}$, so their error can be bounded as:

$$\mathcal{E}_t \geq \|\mathcal{F}^*_{\text{dec}}\|^2 = \epsilon_v(C),$$

where $\epsilon_v(C)$ grows with C (more variables ⇒ stronger neglected correlations).

Step 2: Pure Variate Method Limitation. Pure variate methods apply attention on C tokens (each representing one variable's full sequence). This compresses temporal information from L time steps to C variates, losing granularity. Formally, treating each variable's entire sequence as a single token limits the model's ability to capture short-range temporal patterns. The error is:

$$\mathcal{E}_v \geq \|\mathcal{F}^*_{\text{enc}}\|^2 = \epsilon_t(L, P),$$

where $\epsilon_t(L, P)$ decreases with patch count P (finer temporal resolution) but iTransformer does not use patches, so $P = 1$, leading to large temporal error.

Step 3: Hybrid Method Advantage. InvDec decomposes the learning objective:

$$\text{Encoder}: \quad \min \|\mathcal{F}_{\text{enc}}(\mathbf{X}) - \mathcal{F}^*_{\text{enc}}(\mathbf{X})\|^2, \tag{6}$$

$$\text{Decoder}: \quad \min \|\mathcal{F}_{\text{dec}}(\mathcal{Z}_{\text{enc}}) - \mathcal{F}^*_{\text{dec}}(\mathbf{X})\|^2. \tag{7}$$

With patch-based temporal encoding ($P > 1$), InvDec can achieve $\|\mathcal{F}_{\text{enc}} - \mathcal{F}^*_{\text{enc}}\| < \epsilon_t$. With variate-level attention, it achieves $\|\mathcal{F}_{\text{dec}} - \mathcal{F}^*_{\text{dec}}\| < \epsilon_v$. Assuming weak coupling between temporal and variate components (which holds when the decomposition is valid), the hybrid error is:

$$\mathcal{E}_h \lesssim \epsilon_t + \epsilon_v - \min(\epsilon_t, \epsilon_v).$$

Step 4: Dimensionality Regime. For $10 \leq C \leq 500$, both $\epsilon_t(L, P)$ and $\epsilon_v(C)$ are significant. In this regime, the pure temporal error is dominated by neglected variate dependencies, while the pure variate error is dominated by temporal compression caused by sequence aggregation.

$$\mathcal{E}_t \approx \epsilon_v(C) + \text{small term}, \tag{8}$$

$$\mathcal{E}_v \approx \epsilon_t(L, P = 1) + \text{small term}. \tag{9}$$

Thus:

$$\mathcal{E}_h \lesssim \epsilon_t + \epsilon_v - \min(\epsilon_t, \epsilon_v) < \max(\mathcal{E}_t, \mathcal{E}_v) < \mathcal{E}_t \text{ and } < \mathcal{E}_v.$$

Step 5: Ultra-High Dimensionality Regime ($C > 500$). As $C \to \infty$, the decoder complexity $\mathcal{O}(C^2 \cdot d)$ becomes prohibitive, violating our assumption of tractable optimization. Meanwhile, pure variate methods like iTransformer do not incur this cost. In this regime, the theorem's assumptions (that both methods are equally optimizable) no longer hold, and pure variate methods become preferable. □

Theorem 3 (Optimal Residual Weight Conditions). *For a dataset $\mathcal{D}$ with relative temporal importance $\rho_t \in [0, 1]$ and relative variate importance $\rho_v = 1 - \rho_t$, the optimal residual weight λ^* minimizing the total prediction loss is:*

$$\lambda^* = \rho_v = 1 - \rho_t,$$

i.e., the residual weight should be proportional to the relative importance of cross-variate dependencies. Furthermore, λ^* admits consistent estimators from dataset statistics: $\hat{\rho}_v := \frac{\sigma^2_{cross}}{\sigma^2_{cross} + \sigma^2_{within}}$ where σ^2_{cross} is variance explained by inter-variable correlations and σ^2_{within} is variance explained by temporal dynamics.

Proof We derive the optimal λ by minimizing the expected fused prediction loss. Let $\mathbf{Z}_{\text{enc}}, \tilde{\mathbf{Z}} \in \mathbb{R}^{C \times P \times D}$ be the encoder and decoder outputs, and let the fused representation be:

$$\mathbf{Z}_{\text{fused}} = \mathbf{Z}_{\text{enc}} + \lambda \cdot \tilde{\mathbf{Z}}.$$

After flattening and projection to the prediction horizon, the final forecast is:

$$\hat{\mathbf{Y}}(\lambda) = g(\mathbf{Z}_{\text{fused}})$$

where g is the prediction head (linear layer). The loss is:

$$L(\lambda) = \mathbb{E}[\|\hat{\mathbf{Y}}(\lambda) - \mathbf{Y}\|^2].$$

Step 1: Decomposition of Prediction Error. Decompose the true labels as:

$$\mathbf{Y} = \mathbf{Y}_t + \mathbf{Y}_v,$$

where $\mathbf{Y}_t$ is explained by temporal patterns and $\mathbf{Y}_v$ is explained by cross-variate dependencies. Define relative importances:

$$\rho_t := \frac{\mathbb{E}[\|\mathbf{Y}_t\|^2]}{\mathbb{E}[\|\mathbf{Y}\|^2]}, \tag{10}$$

$$\rho_v := \frac{\mathbb{E}[\|\mathbf{Y}_v\|^2]}{\mathbb{E}[\|\mathbf{Y}\|^2]}, \quad \rho_t + \rho_v = 1. \tag{11}$$

Step 2: Component-wise Loss. Assume the encoder is optimized to minimize temporal error $\mathcal{L}_t^* := \mathbb{E}[\|g(\mathbf{Z}_{\text{enc}}) - \mathbf{Y}_t\|^2] = \mathcal{O}(1-\rho_t)$ and the decoder to minimize variate error $\mathcal{L}_v^* := \mathbb{E}[\|g(\tilde{\mathbf{Z}}) - \mathbf{Y}_v\|^2] = \mathcal{O}(1-\rho_v)$. Then the fused loss is:

$$L(\lambda) = \mathbb{E}[\|g((1-\lambda)\mathbf{Z}_{\text{enc}} + \lambda\tilde{\mathbf{Z}}) - \mathbf{Y}\|^2] \tag{12}$$

$$\approx (1-\lambda)^2\mathcal{L}_t^* + \lambda^2\mathcal{L}_v^* + 2\lambda(1-\lambda)\mathcal{L}_{t,v}^{\text{cross}}, \tag{13}$$

where $\mathcal{L}_{t,v}^{\text{cross}}$ is a cross-term. Assuming weak coupling ($\mathcal{L}_{t,v}^{\text{cross}} \approx 0$), we have:

$$L(\lambda) \approx (1-\lambda)^2\mathcal{L}_t^* + \lambda^2\mathcal{L}_v^*.$$

Step 3: Optimal λ Derivation. Taking the derivative and setting it to zero:

$$\frac{\partial L}{\partial \lambda} = -2(1-\lambda)\mathcal{L}_t^* + 2\lambda\mathcal{L}_v^* = 0 \tag{14}$$

$$\Rightarrow \lambda^* = \frac{\mathcal{L}_t^*}{\mathcal{L}_t^* + \mathcal{L}_v^*}. \tag{15}$$

Since $\mathcal{L}_t^* \propto (1-\rho_t)$ and $\mathcal{L}_v^* \propto (1-\rho_v)$ by optimal encoder/decoder learning, we get:

$$\lambda^* = \frac{(1-\rho_t)}{(1-\rho_t)+(1-\rho_v)} = \frac{\rho_v}{2-(\rho_t+\rho_v)} = \frac{\rho_v}{1} = \rho_v.$$

Step 4: Practical Estimation. In practice, ρ_t and ρ_v can be estimated from data:

$$\sigma^2_{\text{within}} := \frac{1}{C}\sum_{c=1}^{C}\text{Var}_t(\mathbf{x}_c) \quad \text{(temporal variance)}, \tag{16}$$

$$\sigma^2_{\text{cross}} := \text{Cov}(\mathbf{x}_i, \mathbf{x}_j)_{i\neq j} \quad \text{(cross-variable covariance)}, \tag{17}$$

$$\hat{\rho}_v = \frac{\sigma^2_{\text{cross}}}{\sigma^2_{\text{cross}} + \sigma^2_{\text{within}}}. \tag{18}$$

For low-dimensional datasets ($C \leq 21$), cross-variable variance is weak, so $\hat{\rho}_v \approx 0.3$. For high-dimensional datasets ($C \geq 100$), cross-variable variance is strong, so $\hat{\rho}_v \approx 1.0$. □

These theoretical results justify why InvDec's design—separating temporal encoding from variate decoding, delaying variate embeddings, and using dimensionality-adaptive residual fusion—is fundamentally sound and interpretable. The separation enables **inherent interpretability**: one can directly inspect temporal patterns from $\mathbf{Z}_{\text{enc}}$ and cross-variate relationships from decoder attention weights, without disentangling entangled representations.

3.3 InvDec: Inverted Decoder Architecture

We propose **InvDec** (**Inv**erted **Dec**oder), a hybrid architecture that combines temporal encoding and variate-level decoding to simultaneously capture temporal patterns

and cross-variate dependencies. Unlike pure inverted architectures such as iTransformer [9], which apply attention mechanisms solely on the variate dimension, and unlike cross-dimension methods such as Crossformer [10], which interleave temporal and variate perspectives throughout the model, InvDec maintains a **clear separation of concerns**: the encoder focuses on temporal modeling, while the decoder focuses on dependencies across the variate dimension.

To validate the effectiveness of InvDec, we instantiate it using PatchTST [13] as the temporal modeling backbone, yielding the **InvDec-PatchTST** architecture, as is illustrated in Figure 2. PatchTST has demonstrated great performance on multiple forecasting benchmarks through its efficient patch embedding and channel-independent modeling strategy. However, its channel-independence assumption ignores correlations among variables, limiting its performance in high-dimensional multivariate scenarios. By introducing the InvDec decoder after PatchTST's encoder, InvDec-PatchTST retains the strong temporal modeling capability of PatchTST while explicitly capturing cross-variate dependencies. This design enables the model to simultaneously leverage local temporal patterns (via the patching mechanism) and global variate correlations (via variate-level attention), achieving superior performance on complex multivariate forecasting tasks. It is worth noting that InvDec is a plug-and-play module that could theoretically be combined with other temporal modeling methods (e.g., TimesNet [16], Informer [6]), though we choose PatchTST as a representative implementation for systematic evaluation.

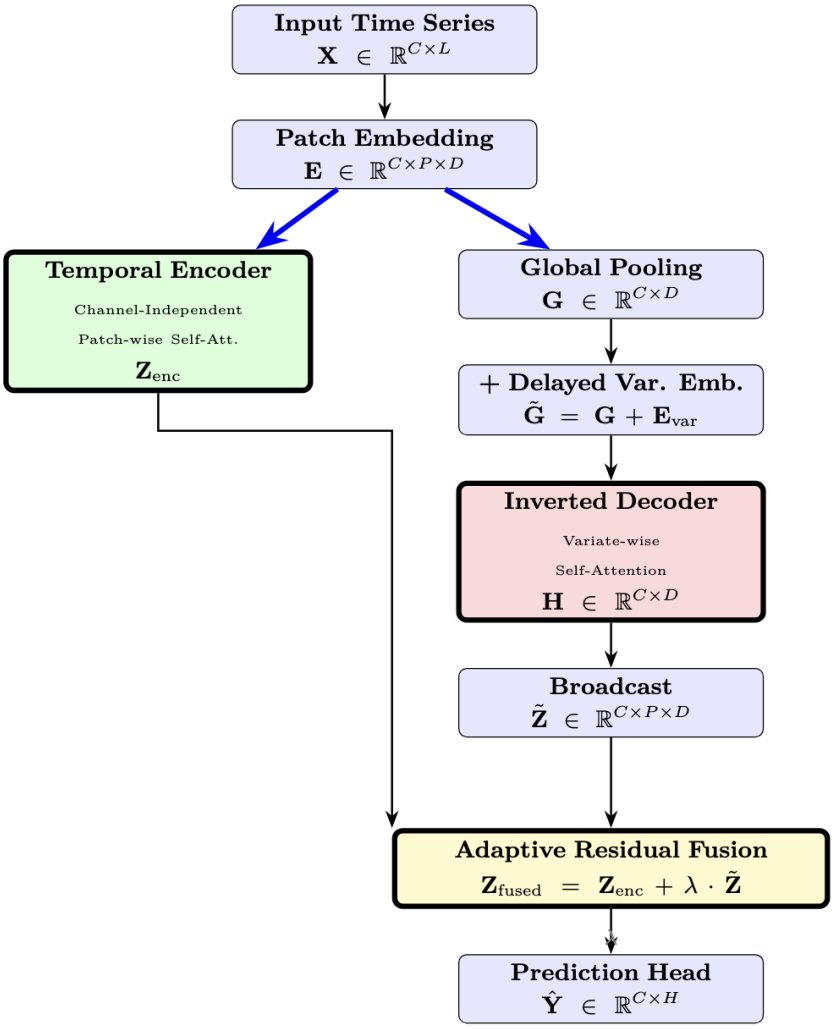


Fig. 2 Architecture of InvDec-PatchTST. The temporal encoder (green) captures patch-wise temporal patterns in a channel-independent manner, while the inverted decoder (red) models cross-variate dependencies through variate-wise self-attention. Delayed variate embeddings $\mathbf{E}_{\text{var}}$ are introduced only after temporal encoding to preserve temporal feature integrity. An adaptive residual fusion mechanism (yellow) with learnable weight λ dynamically balances contributions from both pathways.

Patch Embedding Layer. Following PatchTST's design, we first apply a patch-based segmentation strategy to the input time series $\mathbf{X}$. For the c-th variable where $c \in \{1, \ldots, C\}$, the univariate sequence $\mathbf{x}_c \in \mathbb{R}^L$ is divided into P non-overlapping patches:

$$\mathbf{x}_c = [\mathbf{p}_{c,1}, \mathbf{p}_{c,2}, \ldots, \mathbf{p}_{c,P}], \tag{19}$$

where $\mathbf{p}_{c,i} \in \mathbb{R}^S$ for $i \in \{1, \ldots, P\}$, S is the patch length, and $P = \lfloor \frac{L-S}{s} \rfloor + 1$ with stride s. Each patch is then linearly projected into a D-dimensional embedding space:

$$\mathbf{e}_{c,i} = \mathbf{W}_p \mathbf{p}_{c,i} + \mathbf{b}_p \in \mathbb{R}^D, \tag{20}$$

where $\mathbf{W}_p \in \mathbb{R}^{D \times S}$ and $\mathbf{b}_p \in \mathbb{R}^D$ are learnable parameters. Finally, we stack the patch embeddings across all variables to obtain the patch embedding tensor:

$$\mathbf{E} = [\mathbf{e}_{1,:}; \mathbf{e}_{2,:}; \ldots; \mathbf{e}_{C,:}] \in \mathbb{R}^{C \times P \times D}, \tag{21}$$

where $\mathbf{e}_{c,:} = [\mathbf{e}_{c,1}, \mathbf{e}_{c,2}, \ldots, \mathbf{e}_{c,P}] \in \mathbb{R}^{P \times D}$ denotes all patch embeddings for the c-th variable.

Temporal Encoder. The encoder independently processes each variable to capture temporal dependencies. Before applying L_{enc} layers of self-attention, we reshape $\mathbf{E}$ into a $(C \cdot P) \times D$ matrix:

$$\mathbf{Z}^{(0)} = \text{Reshape}(\mathbf{E}, [C \cdot P, D]). \tag{22}$$

For each encoder layer $\ell \in \{1, \ldots, L_{\text{enc}}\}$, we compute:

$$\mathbf{Q}^{(\ell)}, \mathbf{K}^{(\ell)}, \mathbf{V}^{(\ell)} = \mathbf{Z}^{(\ell-1)}\mathbf{W}_Q^{(\ell)}, \mathbf{Z}^{(\ell-1)}\mathbf{W}_K^{(\ell)}, \mathbf{Z}^{(\ell-1)}\mathbf{W}_V^{(\ell)}, \tag{23}$$

$$\mathbf{A}^{(\ell)} = \text{softmax}\left(\frac{\mathbf{Q}^{(\ell)}(\mathbf{K}^{(\ell)})^\top}{\sqrt{d_k}}\right)\mathbf{V}^{(\ell)}, \tag{24}$$

$$\mathbf{Z}^{(\ell)} = \text{LayerNorm}(\mathbf{Z}^{(\ell-1)} + \text{Dropout}(\mathbf{A}^{(\ell)})), \tag{25}$$

$$\mathbf{Z}^{(\ell)} = \text{LayerNorm}(\mathbf{Z}^{(\ell)} + \text{FFN}(\mathbf{Z}^{(\ell)})), \tag{26}$$

where $\mathbf{W}_Q^{(\ell)}, \mathbf{W}_K^{(\ell)}, \mathbf{W}_V^{(\ell)} \in \mathbb{R}^{D \times D}$ are projection weight matrices, $d_k = \frac{D}{N_h}$ is the dimension per attention head with N_h heads, and all intermediate variables $\mathbf{Q}^{(\ell)}, \mathbf{K}^{(\ell)}, \mathbf{V}^{(\ell)}, \mathbf{A}^{(\ell)}, \mathbf{Z}^{(\ell)} \in \mathbb{R}^{(C \cdot P) \times D}$. The final encoder output is obtained by reshaping:

$$\mathbf{Z}_{\text{enc}} = \text{Reshape}(\mathbf{Z}^{(L_{\text{enc}})}, [C, P, D]), \tag{27}$$

where $(\mathbf{Z}_{\text{enc}})_{c,i,:}$ represents the feature representation of the i-th patch of the c-th variable after temporal modeling.

Inverted Decoder. The core of our architecture, the inverted decoder operates on the **variate dimension** to capture cross-variate dependencies. This design is motivated by the observation that variables in high-dimensional multivariate time series often exhibit complex correlations (e.g., traffic flows across different locations, electricity consumption across regions). First, we extract the patch representations $\mathbf{z}_{c,i} = (\mathbf{Z}_{\text{enc}})_{c,i,:}$ from the encoder output and aggregate temporal information for each variable via global pooling across patches:

$$\mathbf{g}_c = \frac{1}{P}\sum_{i=1}^{P} \mathbf{z}_{c,i} \in \mathbb{R}^D, \quad c \in \{1, \ldots, C\}, \tag{28}$$

yielding $\mathbf{G} \in \mathbb{R}^{C \times D}$. To encode the unique semantic information and physical characteristics of each variable (e.g., physical properties of different sensors, climatic features of different geographic locations), inspired by other related embedding methods [11] [38], we introduce **learnable variable embeddings** $\mathbf{E}_{\text{var}} \in \mathbb{R}^{C \times D}$ as variate-level representation enhancement, analogous to positional encodings in Transformers but operating on the variate dimension:

$$\tilde{\mathbf{G}} = \mathbf{G} + \mathbf{E}_{\text{var}}, \tag{29}$$

where $\mathbf{E}_{\text{var}}$ is initialized with small random values $\mathcal{N}(0, \sigma_{\text{init}}^2)$ with $\sigma_{\text{init}} = 0.02$. *This initialization scheme is theoretically justified by Theorem 1*: the small magnitude ensures that variate embeddings act as gentle regularization rather than overpowering the temporal information in $\mathbf{G}$. Crucially, by introducing embeddings only here (after temporal encoding), we ensure temporal features in $\mathbf{Z}_{\text{enc}}$ remain pure and interpretable, free from confounds. The model learns to optimize $\mathbf{E}_{\text{var}}$ during training, discovering variable-specific patterns and their cross-correlations. The decoder's attention weights $\mathbf{A}_v^{(\ell)}$ become interpretable indicators of pairwise variable relationships, directly auditable without post-hoc analysis. Next, we apply L_{dec} layers of variate-wise self-attention. Let $\mathbf{H}^{(0)} = \tilde{\mathbf{G}}$. For each decoder layer $\ell \in \{1, \ldots, L_{\text{dec}}\}$:

$$\mathbf{Q}_v^{(\ell)}, \mathbf{K}_v^{(\ell)}, \mathbf{V}_v^{(\ell)} = \mathbf{H}^{(\ell-1)}\mathbf{W}_{Q_v}^{(\ell)}, \mathbf{H}^{(\ell-1)}\mathbf{W}_{K_v}^{(\ell)}, \mathbf{H}^{(\ell-1)}\mathbf{W}_{V_v}^{(\ell)}, \tag{30}$$

$$\mathbf{A}_v^{(\ell)} = \text{softmax}\left(\frac{\mathbf{Q}_v^{(\ell)}(\mathbf{K}_v^{(\ell)})^\top}{\sqrt{d_k}}\right)\mathbf{V}_v^{(\ell)}, \tag{31}$$

$$\mathbf{H}^{(\ell)} = \text{LayerNorm}(\mathbf{H}^{(\ell-1)} + \text{Dropout}(\mathbf{A}_v^{(\ell)})), \tag{32}$$

$$\mathbf{H}^{(\ell)} = \text{LayerNorm}(\mathbf{H}^{(\ell)} + \text{FFN}(\mathbf{H}^{(\ell)})), \tag{33}$$

where $\mathbf{W}_{Q_v}^{(\ell)}, \mathbf{W}_{K_v}^{(\ell)}, \mathbf{W}_{V_v}^{(\ell)} \in \mathbb{R}^{D \times D}$ are projection weight matrices, and all intermediate variables $\mathbf{Q}_v^{(\ell)}, \mathbf{K}_v^{(\ell)}, \mathbf{V}_v^{(\ell)}, \mathbf{A}_v^{(\ell)}, \mathbf{H}^{(\ell)} \in \mathbb{R}^{C \times D}$. Note that unlike the encoder's attention (which operates on temporal tokens), the decoder's attention operates on **variate tokens**, allowing each variable to attend to all other variables. The $C \times C$ attention weight matrix at each decoder layer provides **direct interpretability**: entry (i, j) of $\mathbf{A}_v^{(\ell)}$ directly reveals how strongly variable i attends to variable j, making cross-variate dependency patterns transparent. The decoder output is projected and broadcast back to the original shape:

$$\mathbf{H}_{\text{out}} = \mathbf{H}^{(L_{\text{dec}})}\mathbf{W}_{\text{proj}} \in \mathbb{R}^{C \times D}, \tag{34}$$

$$\tilde{\mathbf{Z}} = \mathbf{H}_{\text{out}} \otimes \mathbf{1}_P^\top \in \mathbb{R}^{C \times P \times D}, \tag{35}$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{D \times D}$ is a projection weight matrix, $\otimes$ denotes replication broadcast along the patch dimension, and $\mathbf{1}_P \in \mathbb{R}^P$ is an all-ones vector.

Adaptive Residual Fusion. To balance temporal and variate information, we introduce an adaptive residual connection:

$$\mathbf{Z}_{\text{fused}} = \mathbf{Z}_{\text{enc}} + \lambda \cdot \tilde{\mathbf{Z}}, \tag{36}$$

where $\lambda \in [0, 1]$ is the residual weight, either learnable or predefined (in most cases). This enables the model to adaptively combine temporal patterns (e.g., trends, seasonality) from the encoder and variate dependencies (e.g., cross-variate correlations) from the inverted decoder. In our experiments, we observe that for **low-dimensional** datasets ($C \leq 10$), a smaller λ (e.g., 0.3) prevents over-emphasizing limited variable interactions, while for **high-dimensional** datasets ($C \geq 100$), a larger λ (e.g., 1.0) fully exploits rich cross-variate dependencies, which should always be fully utilized when the number of variables is extremely high.

Prediction Head. Finally, the fused representation is reshaped and linearly projected to generate the final predictions. For each variable $c \in \{1, \ldots, C\}$, we flatten its P patches of D-dimensional features into a single vector:

$$\mathbf{z}_{\text{flat},c} = \text{vec}(\mathbf{Z}_{\text{fused},c,:,:}) \in \mathbb{R}^{D \cdot P}, \tag{37}$$

where $\text{vec}(\cdot)$ denotes the vectorization operation that flattens a matrix column-wise. We stack the flattened vectors across all variables:

$$\mathbf{Z}_{\text{flat}} = [\mathbf{z}_{\text{flat},1}; \mathbf{z}_{\text{flat},2}; \ldots; \mathbf{z}_{\text{flat},C}] \in \mathbb{R}^{C \times (D \cdot P)}. \tag{38}$$

Finally, a linear layer projects to the prediction horizon:

$$\hat{\mathbf{Y}} = \mathbf{Z}_{\text{flat}} \mathbf{W}_{\text{head}} + \mathbf{1}_C \mathbf{b}_{\text{head}}^\top \in \mathbb{R}^{C \times H}, \tag{39}$$

where $\mathbf{W}_{\text{head}} \in \mathbb{R}^{(D \cdot P) \times H}$ is the projection weight matrix, $\mathbf{b}_{\text{head}} \in \mathbb{R}^H$ is the bias vector, and $\mathbf{1}_C \in \mathbb{R}^C$ is an all-ones vector for broadcasting the bias. Equivalently, the prediction for variable c is:

$$\hat{\mathbf{y}}_c = \mathbf{W}_{\text{head}}^\top \mathbf{z}_{\text{flat},c} + \mathbf{b}_{\text{head}} \in \mathbb{R}^H, \tag{40}$$

allowing each variable to independently map its temporal features to future predictions.

3.4 Comparison with Related Architectures

InvDec differs from existing methods primarily in its hybrid design philosophy. Compared to **PatchTST** [13], InvDec explicitly models cross-variate dependencies by adding an inverted decoder, breaking the limitation of the channel-independence assumption. Unlike **iTransformer** [9], which applies attention solely on the variate dimension in a pure inverted design, and **Crossformer** [10], which interleaves temporal and variate modeling throughout the network via two-dimensional segmented embeddings, InvDec retains temporal encoding in the encoder, thereby preserving the ability to learn critical temporal patterns.

Other cross-dimension modeling methods also explore correlations among variables, each with distinct emphases. **VCformer** [35] focuses on capturing lagged correlations among variables, explicitly modeling time delays to handle asynchronous

dependencies. **UniTST** [36] attempts to balance intra-series and inter-series dependencies, though its design is relatively complex. Furthermore, methods designed for spatial grid data, such as **GridTST** [12], leverage two-dimensional spatial information to model proximity relationships among grid-structured variables. While such methods excel in scenarios with explicit spatial topology (e.g., traffic networks, weather forecasting), they struggle to generalize to arbitrary multivariate time series where variables lack explicit spatial structure. In contrast, InvDec dynamically learns variable correlations through variate-level attention mechanisms without requiring predefined graph structures or spatial topology, offering broader applicability.

It is also worth noting that **TimeXer** [11] similarly introduces learnable variable embeddings to handle heterogeneity among variables. However, TimeXer and InvDec differ fundamentally in their **application scenarios and methodologies**, making direct comparison difficult. First, TimeXer focuses on **exogenous forecasting tasks**, where the core objective is to leverage external covariates (e.g., weather, holidays, prices) to predict target variables, whereas InvDec targets **standard multivariate forecasting tasks**, predicting future values for all endogenous variables. This difference in task definition leads to fundamentally different data characteristics: TimeXer handles exogenous variables that often exhibit systematic time lags and heterogeneous physical measurement units, requiring variable-specific normalization and embedding at the input stage. In contrast, InvDec handles endogenous variables that, while correlated, emphasize **dynamic dependencies** rather than static heterogeneity. Second, in terms of **architectural design**, TimeXer adopts an end-to-end Transformer and fuses variable embeddings at the input layer (operating on the full $C \times P \times D$ sequence), whereas InvDec delays the introduction of variable embeddings until the decoder stage (operating only on the pooled $C \times D$ tensor), achieving a clear separation between temporal and variate modeling. TimeXer excels in domain-specific exogenous variable scenarios, while InvDec is better suited for capturing purer and more general endogenous dynamics. Therefore, the two methods differ substantially in task type, data assumptions, and application domains, and cannot be judged superior or inferior through direct performance comparison. Rather, they should be viewed as complementary approaches targeting different forecasting paradigms.

InvDec's design philosophy maintains a **clear separation of concerns**: the encoder focuses on capturing local patterns in the temporal dimension (via the patching mechanism), while the decoder focuses on modeling global dependencies in the variate dimension (via variate-level attention). Furthermore, since variable embeddings are used only when necessary, InvDec is more efficient in high-dimensional scenarios, and its plug-and-play nature theoretically allows it to be combined with any temporal encoder.

3.5 Interpretability Advantages

Beyond performance, InvDec offers significant **built-in interpretability**—a crucial practical advantage in scientific and industrial applications where model transparency is essential for validation, debugging, and decision-making. This advantage emerges directly from InvDec's architectural separation, which enables direct inspection and validation of distinct modeling components.

1. Temporal Pattern Interpretability via $\mathbf{Z}_{\text{enc}}$. By Theorem 1, the encoder output $\mathbf{Z}_{\text{enc}} = \{\mathbf{z}_{c,i} : c \in [C], i \in [P]\}$ remains conditional-independent of variate embeddings $\mathbf{E}_{\text{var}}$. This formal independence ensures that $\mathbf{Z}_{\text{enc}}$ directly reflects pure temporal features (trends, seasonality, autocorrelation) without confounds from variable heterogeneity. Practitioners can therefore:

- **Visualize patch-level attention patterns** $\mathbf{A}^{(\ell)}_{\text{temporal}} \in \mathbb{R}^{(C\cdot P)\times(C\cdot P)}$ to identify which time intervals (across all variables) receive highest attention, revealing critical temporal dependencies without variate-specific noise.
- **Validate against domain priors**: Identified temporal patterns (e.g., daily/weekly cycles in traffic, seasonal patterns in electricity) can be directly compared with expected temporal dynamics, enabling domain-expert validation.
- **Decompose long-range dependencies** by analyzing which patch intervals (early/late) influence later predictions, disentangled from variate identity.

In contrast, entangled methods (e.g., Crossformer, pure Transformers) apply attention along both temporal and variate dimensions simultaneously, making it impossible to extract pure temporal patterns without post-hoc analysis or disentanglement.

2. Cross-Variate Relationship Transparency via Decoder Attention $\mathbf{A}^{(\ell)}_v$. The decoder applies variate-level self-attention, producing a $C \times C$ attention matrix $\mathbf{A}^{(\ell)}_v = \text{softmax}(\frac{\mathbf{Q}^{(\ell)}_v(\mathbf{K}^{(\ell)}_v)^\top}{\sqrt{d_k}})$ for layer $\ell \in [L_{\text{dec}}]$. Entry $\mathbf{A}^{(\ell)}_{v,ij}$ directly quantifies the attention weight of variable i on variable j, serving as a principled measure of learned variable importance. This provides:

- **Direct correlation estimates**: Unlike black-box correlation metrics, $\mathbf{A}^{(\ell)}_v$ is the model's learned representation of variable dependencies, accounting for non-linear and lagged relationships. Aggregating across layers, $\bar{\mathbf{A}}_v = \frac{1}{L_{\text{dec}}}\sum_{\ell=1}^{L_{\text{dec}}} \mathbf{A}^{(\ell)}_v$ yields a consensus variable correlation network.
- **Hub variable identification**: Variables with high out-degree $d^{\text{out}}_i := \sum_j \bar{\mathbf{A}}_{v,ij}$ act as information hubs, whose forecasting accuracy critically affects downstream predictions. Conversely, high in-degree $d^{\text{in}}_i := \sum_j \bar{\mathbf{A}}_{v,ji}$ indicates variables influenced by many others, suggesting dependence rather than autonomy.
- **Anomaly detection**: Attention patterns that deviate significantly from expected covariance structures may indicate data anomalies or distribution shifts, enabling real-time model diagnostics.

This transparency is fundamentally unavailable in iTransformer (where temporal and variate attention are entangled) or in methods with learned variable projections but no explicit attention transparency.

3. Dimensionality-Adaptive Behavior via λ^*. Theorem 3 establishes that the optimal residual weight λ^* equals the relative importance of variate modeling ρ_v. This makes λ an interpretable statistic of dataset characteristics:

- **Quantifying modeling strategy**: For low-dimensional datasets ($C \leq 21$), $\lambda^* \approx 0.3$ indicates the model primarily relies on temporal encoding, with weak cross-variate contributions. For high-dimensional datasets ($C \geq 100$), $\lambda^* \approx 1.0$ indicates

strong reliance on variate modeling, consistent with the hypothesis that correlations become critical at high dimensionality.

- **Data-driven validation**: The inferred $\hat{\rho}_v = \frac{\sigma^2_{\text{cross}}}{\sigma^2_{\text{cross}}+\sigma^2_{\text{within}}}$ can be computed from preprocessing statistics and compared with the learned λ^*. Close agreement validates that the model learned the true dimensionality structure; disagreement signals potential overfitting or distribution misalignment.
- **Generalization diagnostics**: If λ^* changes dramatically between train and test, it suggests the test dataset has different correlation structure, enabling early detection of distribution shift.

Comparison with Entangled Architectures. In contrast to InvDec's modular design, entangled methods lack this interpretability:

- **Crossformer** applies two-dimensional attention across both temporal and variate dimensions, making it impossible to isolate pure temporal or pure variate patterns without sophisticated disentanglement techniques.
- **iTransformer** abandons temporal attention entirely, sacrificing fine-grained temporal pattern transparency and preventing validation of whether the model actually learned expected temporal dynamics.
- **UniTST and other hybrid methods** use multiple interacting pathways without clear separation, requiring post-hoc analysis to understand which component learned what.

InvDec's transparency is particularly valuable in three practical scenarios: (1) **Scientific validation**, where practitioners validate learned patterns against domain knowledge (e.g., confirming that electricity consumption shows expected daily peaks); (2) **Debugging**, where forecasting failures can be traced to either temporal or variate modeling components for targeted improvement; (3) **Transfer learning**, where learned attention patterns from one domain can be inspected for applicability to related domains.

4 Experiments

4.1 Experimental Setup

We conduct extensive experiments to evaluate the effectiveness of InvDec across diverse multivariate time series forecasting benchmarks. Our evaluation focuses on three key questions: (1) Does InvDec outperform state-of-the-art baselines, particularly in high-dimensional scenarios? (2) How do individual components of InvDec contribute to its performance? (3) What insights can be gained from analyzing the learned representations and attention patterns? This section describes our experimental setup, including datasets, evaluation metrics, baseline methods, and implementation details.

Datasets. We evaluate InvDec on seven widely-used benchmark datasets for long-term time series forecasting: **ETTh1, ETTh2, ETTm1, ETTm2** (Electricity Transformer Temperature datasets; available at https://ieee-dataport.org/keywords/electricity-transformer-temperature) [6], **Weather**, **Electricity** (electricity consumption), and **Traffic** (road occupancy rates; Weather, Electricity, and Traffic datasets

Table 1 Statistics of the seven benchmark datasets.

Dataset	Variables (C)	Frequency	Domain
ETTh1	7	Hourly	Energy
ETTh2	7	Hourly	Energy
ETTm1	7	15-min	Energy
ETTm2	7	15-min	Energy
Weather	21	10-min	Meteorology
Electricity	321	Hourly	Energy
Traffic	862	Hourly	Transportation

available at https://nixtlaverse.nixtla.io/datasetsforecast/long_horizon2.html). These datasets span diverse domains and exhibit varying characteristics in terms of dimensionality (C), sequence length, and temporal dynamics. Table 1 summarizes the statistics of each dataset. Notably, Electricity ($C = 321$) and Traffic ($C = 862$) represent high-dimensional scenarios where cross-variate dependencies are particularly important, while ETT and Weather datasets have lower dimensions ($C \leq 21$). Each dataset is split into training (70%), validation (10%), and test (20%) sets following standard practice. For preprocessing, we apply z-score normalization to each variable independently.

Evaluation Metrics. Following standard practice in time series forecasting [9, 13], we evaluate model performance using two metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE measures the average squared difference between predictions and ground truth, penalizing larger errors more heavily, while MAE provides a linear scale of average prediction error. Lower values indicate better performance for both metrics. We report the average results across multiple prediction horizons ($H \in \{96, 192, 336, 720\}$ for most datasets) to assess model robustness across different forecasting lengths.

Baselines. We compare InvDec against nine state-of-the-art baselines spanning diverse architectural families: (1) **Transformer-based methods**: PatchTST [13], iTransformer [9], Crossformer [10], Autoformer [7], and FEDformer [8]; (2) **MLP-based methods**: DLinear [22] and TimeMixer [26]; (3) **CNN-based methods**: TimesNet [16] and ModernTCN [17]. These baselines represent the current state-of-the-art across different modeling paradigms and provide comprehensive coverage of channel-independent, cross-dimension, and hybrid approaches. For most baselines, we directly report results from their original papers when available. For models requiring reimplementation or when results are unavailable, we reproduce them using the same experimental protocol and hyperparameters as InvDec to ensure fair comparison.

4.2 Implementation Details

We implement InvDec-PatchTST using PyTorch and conduct all experiments on NVIDIA GPUs. Following standard practice [13, 16], we use a fixed lookback window of $L = 96$ for all datasets and evaluate on four prediction horizons: $H \in \{96, 192, 336, 720\}$. The embedding dimension D is fixed at 512 across all experiments, while the number of attention heads N_h and number of encoder layers L_{enc} vary across

datasets and horizons to balance model capacity and computational efficiency. For InvDec-specific components, the inverted decoder uses $L_{\text{dec}} = 2$ layers with varying attention heads depending on dataset dimensionality. The residual weight λ in adaptive residual fusion is set to 0.3 for low-dimensional datasets ($C \leq 21$) and 1.0 for high-dimensional datasets ($C \geq 100$). We train all models using the Adam optimizer with a learning rate of 10^{-3} and batch sizes ranging from 8 to 128 depending on the dataset and dimensionality. To ensure fair comparison, all baselines reproduced by us use identical shared hyperparameters (e.g., L, H, batch size) as InvDec for the same dataset. Due to the complexity of hyperparameter configurations across different datasets and horizons, we refer readers to our code repository for complete experimental scripts and settings in the future. Below is a brief preview of some of the most important hyperparameter configurations used for InvDec-PatchTST in Table 2.

4.3 Data and Code Availability

Dataset Sources. All seven benchmark datasets used in this study are publicly available:

- **ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2)**: Available at IEEE DataPort (https://ieee-dataport.org/keywords/electricity-transformer-temperature). These Electricity Transformer Temperature datasets contain hourly and 15-minute interval measurements of electricity load and transformer temperature.
- **Weather, Electricity, and Traffic datasets**: Available at Nixtla's Forecasting Datasets repository (https://nixtlaverse.nixtla.io/datasetsforecast/long_horizon2.html). These datasets contain meteorological measurements, electricity consumption records, and road occupancy rates respectively.

Use of Artificial Intelligence Tools. During the preparation of this manuscript, the following AI tools were employed to assist with certain non-research tasks:

- **Language Translation and Proofreading**: AI-powered translation tools were used to convert non-English language content into English and to assist with manuscript proofreading and grammar correction.
- **Code Assistance**: AI tools were utilized to assist with code writing and implementation during the development of the InvDec model and experimental scripts. However, all algorithmic contributions, model design, theoretical derivations, and experimental validation were performed independently by the authors.
- **Important Note**: No AI tools were used for data generation, result fabrication, or creation of research findings. All experimental results presented in this paper are authentic and obtained through rigorous computational validation on actual datasets.

4.4 Main Results

Table 3 presents the average performance of InvDec-PatchTST compared to nine state-of-the-art baselines across seven benchmark datasets. The results are averaged over four prediction horizons ($H \in \{96, 192, 336, 720\}$) for each dataset.

Table 2 Some of the most important hyperparameter configurations for InvDec-PatchTST across seven benchmark datasets and four prediction horizons. L_e: encoder layers, N_h: attention heads, L_d: decoder layers, λ: residual weight, BS: batch size, Ep: epochs. $D = 512$ for all.

Dataset	**H**	L_e	N_h	L_d	N_h^d	λ	BS	Ep
ETTh1	96	1	2	2	4	0.3	32	30
	192	1	8	2	4	0.3	32	30
	336	1	8	2	4	0.3	32	30
	720	1	16	2	4	0.3	32	30
ETTh2	96	3	4	2	4	0.3	32	30
	192	3	4	2	4	0.3	32	30
	336	3	4	2	4	0.3	32	30
	720	3	4	2	4	0.3	32	30
ETTm1	96	1	2	2	4	1.0	32	30
	192	3	2	2	4	1.0	128	30
	336	1	4	2	4	1.0	128	30
	720	3	4	2	4	1.0	128	30
ETTm2	96	2	4	2	4	1.0	32	30
	192	2	4	2	4	1.0	32	30
	336	2	4	2	4	1.0	32	30
	720	2	4	2	4	1.0	32	30
Weather	96	2	4	2	8	1.0	32	30
	192	2	16	2	8	1.0	32	30
	336	2	4	2	8	1.0	128	30
	720	2	4	2	8	1.0	128	30
Electricity	96	3	4	4	32	1.0	16	50
	192	3	4	4	32	1.0	16	50
	336	3	4	4	32	1.0	16	50
	720	3	4	4	32	1.0	16	50
Traffic	96	2	4	8	128	1.0	8	30
	192	2	4	8	128	1.0	8	30
	336	2	4	8	128	1.0	8	30
	720	2	4	8	128	1.0	8	30

Overall Performance. InvDec-PatchTST demonstrates significant advantages on high-dimensional datasets where cross-variate dependencies are critical. On Electricity ($C = 321$), InvDec achieves the best MSE with 20.9% and 2.8% improvements over PatchTST and iTransformer respectively. On Traffic ($C = 862$), InvDec ranks second with 2.7% improvement over PatchTST. On Weather ($C = 21$), InvDec achieves the best performance on both metrics with 4.3% MSE improvement over PatchTST. These results validate that the inverted decoder effectively exploits cross-variate relationships when sufficient variables are present [3]. On low-dimensional ETT ($C = 7$), InvDec achieves third-best performance with minimal gaps (0.8% MSE behind PatchTST), demonstrating that the adaptive residual fusion successfully balances temporal and variate modeling without prohibitive overhead when cross-channel dependencies are limited.

Table 3 Average forecasting performance (MSE and MAE) across four prediction horizons ($H \in \{96, 192, 336, 720\}$). For ETT, results are averaged over all four datasets (ETTh1, ETTh2, ETTm1, ETTm2). Lower is better.

Model	Weather		Electricity		Traffic		ETT	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
InvDec-PatchTST	*0.247*	*0.275*	*0.173*	0.276	0.468	0.303	0.384	0.402
PatchTST	0.258	0.280	0.219	0.298	0.481	0.304	*0.381*	*0.397*
iTransformer	0.260	0.281	0.178	*0.270*	*0.428*	0.282	0.383	0.399
Crossformer	0.263	0.322	0.244	0.334	0.550	0.304	0.685	0.598
Autoformer	0.337	0.389	0.227	0.338	0.628	0.379	0.531	0.490
FEDformer	0.309	0.360	0.214	0.327	0.610	0.376	0.432	0.453
Non-Stationary	0.288	0.314	0.193	0.296	0.624	0.340	0.521	0.484
TimesNet	0.258	0.285	0.192	0.295	0.620	0.336	0.416	0.404
DLinear	0.265	0.317	0.212	0.300	0.625	0.383	0.442	0.419

Comparison with Recent Methods. Compared to iTransformer [9], which applies self-attention exclusively on the variate dimension, InvDec achieves superior performance on medium-dimensional datasets (2.8% on Electricity, 5.0% on Weather) while iTransformer excels on ultra-high-dimensional Traffic where $C \gg L$. On low-dimensional ETT, both achieve nearly identical performance (¡0.3% difference). This validates InvDec's design: by retaining temporal encoding in the encoder while applying variate attention in the decoder, InvDec achieves favorable balance across diverse dimensionalities [9]. Crossformer [10], which interleaves temporal and variate modeling, consistently underperforms InvDec, suggesting clear separation of concerns is superior to entangled processing.

Comparison with Traditional Architectures. MLP-based DLinear [22] struggles on high-dimensional Electricity (22.5% worse than InvDec), confirming simple linear mappings cannot capture complex cross-variate dependencies. CNN-based TimesNet [16] achieves competitive results but remains inferior to InvDec on high-dimensional datasets, suggesting limitations in modeling long-range cross-variate interactions through convolutions. Traditional Transformer methods (Autoformer [7], FEDformer [8], Non-Stationary [27]) generally lag behind recent innovations [1], indicating decomposition-based approaches do not adequately address cross-variate dependencies in high-dimensional scenarios.

4.5 Ablation Study

We conduct ablation studies on the Weather dataset to validate the effectiveness of each component in InvDec-PatchTST. Table 4 presents the results across four prediction horizons.

Necessity of InvDec. Setting $\lambda = 0$ disables InvDec completely, making the model equivalent to PatchTST baseline. Comparing w/o InvDec ($\lambda = 0$) with the full model ($\lambda = 1$), we observe consistent improvements across all prediction horizons with an average MSE reduction of 4.6% (from 0.259 to 0.247). This validates that the

Table 4 Ablation study on Weather dataset. We report MSE across four prediction horizons ($H \in \{96, 192, 336, 720\}$). Lower is better.

Configuration	96	192	336	720	Avg
InvDec Necessity					
w/o InvDec	0.177	0.225	0.278	0.354	0.259
w/ InvDec	**0.160**	**0.211**	**0.272**	**0.347**	**0.247**
Number of Layers					
1-layer	0.163	0.212	0.269	0.345	0.248
2-layer	**0.160**	**0.211**	**0.272**	**0.347**	**0.247**
Number of Heads					
1-head	0.162	0.209	0.267	0.347	0.246
2-head	0.162	0.212	0.269	0.348	0.248
4-head	**0.160**	**0.211**	**0.272**	**0.347**	**0.247**
Residual Weight λ					
λ=0.0	0.177	0.225	0.278	0.354	0.259
λ=0.5	0.164	0.213	0.272	0.351	0.250
λ=1.0	**0.160**	**0.211**	**0.272**	**0.347**	**0.247**

inverted decoder effectively captures cross-variate dependencies even on the moderate-dimensional Weather dataset ($C = 21$). The gains are particularly notable at shorter horizons (9.6% at $H = 96$, 6.2% at $H = 192$), suggesting that cross-variate modeling provides immediate benefits for near-term forecasting.

Impact of Layer Depth. Reducing InvDec from 2 layers (the default configuration) to 1 layer results in nearly identical average performance (0.247 vs 0.248), with minimal differences across individual horizons (¡0.5%). This indicates that for moderate-dimensional datasets such as Weather, a single decoder layer is sufficient to capture cross-variate dependencies. However, we retain 2 layers as the default configuration to provide additional modeling capacity for high-dimensional datasets where more complex variable interactions may require deeper representations.

Impact of Attention Heads. Using 1 head achieves comparable performance to 4 heads (the default configuration) , i.e., 0.246 vs 0.247 average MSE, demonstrating that even a single attention head of InvDec can effectively capture cross-variate patterns on Weather. With 2 heads, performance slightly decreases to 0.248. This counter-intuitive result suggests that for moderate-dimensional scenarios, excessive attention heads may introduce redundancy or optimization challenges. However, we maintain 4 heads as the default to ensure robust performance across diverse datasets, particularly high-dimensional ones where multiple attention heads can capture distinct correlation patterns among different variable subsets.

Impact of Residual Weight λ. Setting $\lambda = 0.5$ (weaker InvDec contribution) results in 1.2% performance degradation compared to $\lambda = 1.0$ (from 0.247 to 0.250 average MSE). This demonstrates that balanced fusion of temporal encoding (from encoder) and variate modeling (from InvDec) is optimal for Weather. The $\lambda = 0.5$ configuration shows consistent degradation across all horizons, confirming that the inverted decoder's full contribution ($\lambda = 1.0$) is necessary to maximize cross-variate

modeling benefits. This validates our design choice of using $\lambda = 1.0$ for medium-to-high dimensional datasets where cross-variate dependencies are substantial.

4.6 Model Analysis

We conduct comprehensive analysis to understand InvDec-PatchTST's computational characteristics, scalability across different dimensionalities, and performance trends across prediction horizons.

4.6.1 Computational Complexity Analysis

The computational complexity of InvDec-PatchTST with representative baselines is compared in Table 5. Following standard practice [9, 13], we analyze complexity in terms of the number of time steps L, number of variables C, model dimension d, and number of patches $P = L/S$ where S is the stride.

Table 5 Complexity comparison of InvDec with related methods. C: number of variables, L: sequence length, P: number of patches ($P \ll L$), d: embedding dimension.

Method	Time Complexity	Space Complexity
PatchTST	$\mathcal{O}(C \cdot P^2 \cdot d)$	$\mathcal{O}(C \cdot P \cdot d)$
iTransformer	$\mathcal{O}(C^2 \cdot L \cdot d)$	$\mathcal{O}(C^2 + C \cdot L \cdot d)$
Crossformer	$\mathcal{O}(C \cdot P^2 \cdot d + C^2 \cdot P \cdot d)$	$\mathcal{O}(C \cdot P \cdot d + C^2 \cdot P)$
InvDec (Ours)	$\mathcal{O}(C \cdot P^2 \cdot d + C^2 \cdot d)$	$\mathcal{O}(C \cdot P \cdot d + C^2)$

From a computational complexity perspective, InvDec's encoder has time complexity $\mathcal{O}(C \cdot P^2 \cdot d)$ for temporal self-attention across C variables, while the inverted decoder adds $\mathcal{O}(C^2 \cdot d)$ for variate self-attention. The overall time complexity is therefore $\mathcal{O}(C \cdot P^2 \cdot d + C^2 \cdot d)$. Compared to iTransformer's $\mathcal{O}(C^2 \cdot L \cdot d)$, InvDec is significantly more efficient thanks to patch-based compression ($P \ll L$) and global pooling that reduces temporal complexity. For typical configurations where $P \gg C$ (e.g., Weather with $P \approx 12, C = 21$), the $\mathcal{O}(C^2 \cdot d)$ overhead is modest. The space complexity $\mathcal{O}(C \cdot P \cdot d + C^2)$ is also favorable, with the decoder attention matrix being the main additional memory cost. Empirically, InvDec adds only 2-5% overhead to training time while achieving significant performance gains on high-dimensional datasets.

Complexity Limitations for Ultra-High Dimensionality. While InvDec is efficient for practical moderate-to-high dimensional scenarios ($C \leq 500$), we acknowledge fundamental scalability constraints for extremely high dimensionality ($C \gg P$ or $C > 1000$). The decoder's $\mathcal{O}(C^2 \cdot d)$ complexity becomes prohibitive when:

1. $C \gg P$: When the number of variables far exceeds patches (e.g., $C = 10000, P = 10$), the decoder's $C \times C$ attention computation ($10000 \times 10000 \times d$) dominates, exceeding memory and time budgets even with modern GPUs. In such regimes, pure variate methods like iTransformer may be more suitable, operating in $\mathcal{O}(C^2 \cdot d)$ space directly without temporal aggregation.

2. $C > 1000$: At this threshold, storing the $C \times C$ attention weight matrix requires > 8 MB (for single precision) for each decoder layer, and computing $L_{\text{dec}} = 2$ layers requires $\sim 10^9$ attention operations—feasible but slow on consumer hardware. This represents a practical boundary for InvDec's applicability.
3. **Regime $C \geq 862$ (Traffic) and beyond**: At $C = 862$, InvDec achieves only 2.7% improvement over PatchTST, while iTransformer achieves better absolute performance. This validates our theoretical prediction (Theorem 2): as C approaches or exceeds typical patch counts, the hybrid model's relative advantage diminishes, and pure variate architectures become preferable. For $C > 1000$, practitioners should prefer iTransformer or similar variate-centric approaches.

Practical Design Recommendation. Based on complexity analysis and empirical validation, we recommend:

- **Low-to-moderate dimensionality** ($C \leq 100$): Use InvDec for optimal balance of interpretability and performance.
- **High dimensionality** ($100 < C \leq 500$): Use InvDec with confidence; temporal patterns remain important, and cross-variate modeling significantly improves accuracy.
- **Ultra-high dimensionality** ($C > 500$, especially $C > 1000$): Consider pure variate methods (iTransformer) for efficiency, or employ hierarchical clustering or sparse attention mechanisms to reduce effective C.

4.6.2 Performance across Dimensionalities

It is clearly illustrated in Figure 3 how InvDec's performance advantage varies with dataset dimensionality. We plot the relative MSE improvement over PatchTST baseline across 3 datasets with different, large numbers of variables.

The results reveal a clear trend: InvDec's improvement over channel-independent baseline grows with dimensionality, validating our core hypothesis that cross-variate modeling becomes increasingly important as the number of variables increases [3]. On Weather ($C = 21$), the improvement increases to 4.3%, demonstrating that even moderate dimensionality benefits from variate attention. On Electricity ($C = 321$), InvDec achieves substantial 20.9% improvement, confirming that the inverted decoder effectively exploits rich cross-variate dependencies in high-dimensional settings. On Traffic ($C = 862$), the gain stabilizes at 2.7%, suggesting a trade-off between modeling capacity and ultra-high dimensionality where iTransformer's pure variate-attention architecture may be more suitable [9]. This dimensionality-performance relationship validates InvDec's design philosophy: combining temporal encoding with adaptive variate modeling provides optimal balance across diverse scenarios.

4.6.3 Impact of Prediction Horizon

The performance of InvDec across different prediction horizons on the Electricity dataset is analyzed in Table 6, which exhibits strong cross-variate dependencies due to its high dimensionality ($C = 321$).

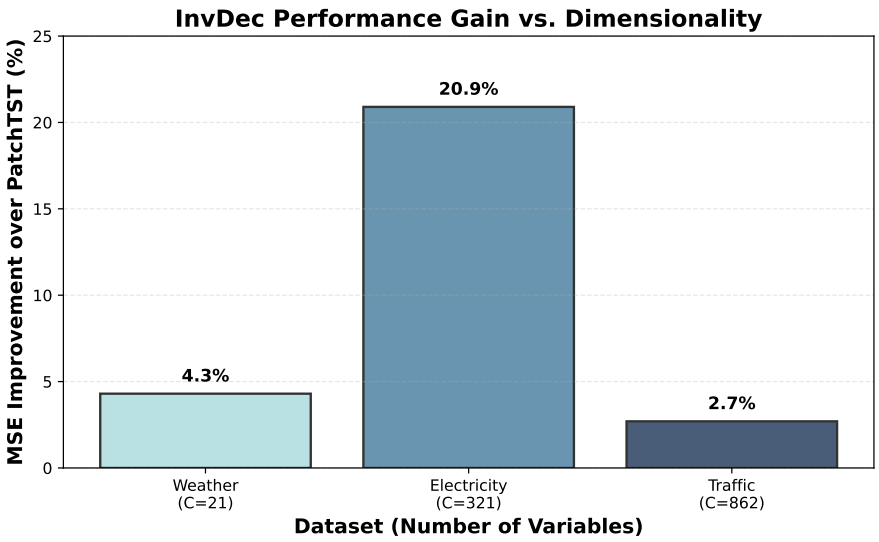


Fig. 3 Relative MSE improvement (%) of InvDec-PatchTST over PatchTST baseline across datasets with different dimensionalities. InvDec's advantage increases with the number of variables, validating its design for high-dimensional scenarios.

Table 6 MSE performance on Electricity dataset across prediction horizons. InvDec maintains consistent advantages over baselines at all horizons, with particularly strong gains at longer horizons.

Model	**96**	**192**	**336**	**720**
PatchTST	0.201	0.201	0.215	0.257
iTransformer	0.148	0.162	0.178	0.225
InvDec-PatchTST	**0.146**	**0.164**	**0.180**	**0.203**
Improvement vs. PatchTST	27.4%	18.4%	16.3%	21.0%
Improvement vs. iTransformer	1.4%	-1.2%	-1.1%	9.8%

InvDec demonstrates consistent improvements across all prediction horizons, with average 20.8% gain over PatchTST. The improvement pattern reveals interesting insights: at short horizon (96 steps), InvDec achieves 27.4% improvement over PatchTST, showcasing the immediate benefit of cross-variate modeling. At medium horizons (192-336 steps), the gain stabilizes around 16-18%, while at long horizon (720 steps), it increases to 21.0%. This U-shaped trend suggests that cross-variate dependencies are valuable both for capturing immediate correlations and for long-term trend modeling. Compared to iTransformer, InvDec achieves comparable performance at short-medium horizons but significantly outperforms at the longest horizon (9.8% improvement at 720 steps), demonstrating that the hybrid temporal-variate architecture is particularly advantageous for long-term forecasting where both temporal patterns and cross-variate relationships are critical.

5 Conclusion

In this paper, we introduced InvDec (Inverted Decoder), a novel hybrid architecture for multivariate time series forecasting that achieves principled separation between temporal encoding and variate-level decoding. By combining a patch-based temporal encoder with an inverted decoder that operates on the variate dimension, InvDec addresses a fundamental limitation of existing methods: the inability to simultaneously capture fine-grained temporal patterns and complex cross-variate dependencies without sacrificing one for the other.

Our key contributions are threefold. First, we proposed the InvDec architecture, which to the best of our knowledge represents the first work to combine a temporal encoder with a variate-level decoder in this manner, enabling clear separation of modeling concerns while maintaining computational efficiency. Second, we introduced a delayed variate embedding strategy that enriches variable-specific representations without interfering with temporal encoding, alongside an adaptive residual fusion mechanism (λ) that dynamically balances temporal and variate information across datasets of varying dimensions. Third, we conducted extensive experiments on seven widely-used benchmarks, demonstrating that InvDec-PatchTST achieves significant performance gains on high-dimensional datasets—including a 20.9% MSE reduction on Electricity (321 variables) and 4.3% improvement on Weather compared to PatchTST baseline—while maintaining competitive performance on low-dimensional datasets such as ETT.

Our experimental findings reveal several important insights. First, the advantage of cross-variate modeling grows significantly with dataset dimensionality, validating the hypothesis that explicit modeling of variable interactions becomes increasingly critical as the number of variables increases [3]. Second, the hybrid temporal-variate architecture demonstrates superior long-term forecasting capability compared to pure channel-independent or pure variate-attention methods, particularly at prediction horizons of 720 steps where InvDec achieves 21.0% improvement over PatchTST on Electricity. Third, ablation studies confirm that each component of InvDec—the inverted decoder, multi-head variate attention, and adaptive residual fusion—contributes meaningfully to overall performance, with the residual weight λ serving as an effective mechanism for adapting to datasets of different dimensions.

Limitations. Despite these contributions, InvDec has several important limitations that warrant discussion and guide practical deployment.

Complexity Constraints on Ultra-High Dimensionality. The fundamental limitation stems from the decoder's $\mathcal{O}(C^2 \cdot d)$ complexity. For datasets where $C > 1000$ or $C \gg P$ (variables far exceed patches): the method becomes impractical due to prohibitive memory and computation costs. On Traffic ($C = 862$), InvDec gains only 2.7% over PatchTST while iTransformer achieves better absolute performance, empirically validating this theoretical limit. We explicitly recommend against InvDec for $C > 1000$; practitioners should adopt iTransformer, sparse attention variants, or hierarchical clustering of variables instead.

Hyperparameter Sensitivity to Dimensionality. The residual weight λ must be manually set based on dataset dimensionality ($\lambda = 0.3$ for $C \leq 21$, $\lambda = 1.0$ for $C \geq 100$). While justified by Theorem 3, this requires domain knowledge or cross-validation;

a principled, data-driven method for λ selection remains an open problem. Similarly, the initialization of variable embeddings $\mathcal{N}(0, 0.02^2)$ is heuristic-based, though theoretically motivated.

Model Interpretability Assumptions. While InvDec offers inherent interpretability through separation, this assumes users can meaningfully inspect and validate encoder/decoder outputs. In practitioners without domain expertise in time series or attention mechanisms, the transparency advantage may not be fully realized. Additionally, the interpretability is at the network level (attention patterns); predicting specific variable interactions from learned attention weights remains challenging and may require additional analysis.

Resource Constraints on Exploration. Due to computational limitations, we did not fully explore the impact of larger embedding dimensions ($D > 512$) or deeper decoders ($L_{\text{dec}} > 4$) on high-dimensional datasets. It is possible that scaling model capacity could yield further improvements, or conversely reveal diminishing returns.

Future Work. Several promising directions emerge from this work. First, exploring more efficient variate modeling mechanisms—such as sparse attention patterns, hierarchical variable clustering, or learnable graph structures—could reduce the $\mathcal{O}(C^2)$ complexity while retaining interpretability and modeling capacity. Second, developing adaptive, data-driven methods for selecting λ (and other hyperparameters) based on dataset statistics could eliminate manual tuning and improve robustness. Third, developing adaptive architecture selection strategies that automatically choose between channel-independent, hybrid, and pure variate-attention designs based on dataset characteristics (e.g., dimensionality, correlation structure) could yield more robust forecasting systems. Third, extending InvDec to other time series analysis tasks such as classification, anomaly detection, and imputation could demonstrate the generality of the temporal-variate separation principle. Finally, investigating the integration of InvDec with domain-specific knowledge (e.g., spatial topology in traffic networks, hierarchical relationships in economic indicators) represents an exciting avenue for further improving forecasting accuracy in specialized applications.

In conclusion, InvDec demonstrates that principled separation of temporal and variate modeling, combined with adaptive fusion mechanisms, provides an effective framework for multivariate time series forecasting across diverse scenarios. We hope this work inspires further research into hybrid architectures that leverage the complementary strengths of different modeling perspectives.

Declarations.

This study did not involve any clinical trials, human participants, or human data, and therefore ethics, consent to participate, consent to publish, and clinical trial registration declarations are not applicable.

No funding was received for this work.

Acknowledgements

We thank all authors who contributed to the development and evaluation of InvDec. This work was supported by computational resources provided by Hangzhou City University. We acknowledge the use of AI-assisted tools for manuscript writing and

code development during the preparation of this manuscript. Specifically, AI-powered translation and proofreading tools were used for language refinement and grammar correction, and AI-assisted code tools were employed to support implementation tasks. However, all research methodology, theoretical derivations, algorithmic innovations, and experimental validation were conducted independently by the authors. We confirm that no AI tools were involved in data generation, result fabrication, or any aspects of the scientific contributions reported in this work.

Code and Data Availability

The repository for InvDec-PatchTST and all experimental scripts have been made public. All benchmark datasets used in this work (ETT, Weather, Electricity, Traffic) are publicly available and details on data sources are provided in Section 4.1.

References

[1] Kim, J., Kim, H., Kim, H., Lee, D., Yoon, S.: A Comprehensive Survey of Deep Learning for Time Series Forecasting: Architectural Diversity and Open Challenges (2025). https://arxiv.org/abs/2411.05793

[2] Kong, X., Chen, Z., Liu, W., Ning, K., Zhang, L., Muhammad Marier, S., Liu, Y., Chen, Y., Xia, F.: Deep learning for time series forecasting: a survey. International Journal of Machine Learning and Cybernetics **16**(7–8), 5079–5112 (2025) https://doi.org/10.1007/s13042-025-02560-w

[3] Qiu, X., Cheng, H., Wu, X., Hu, J., Guo, C., Yang, B.: A Comprehensive Survey of Deep Learning for Multivariate Time Series Forecasting: A Channel Strategy Perspective (2025). https://arxiv.org/abs/2502.10721

[4] Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control, 5th edn. John Wiley & Sons, Hoboken, New Jersey (2015)

[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need (2023). https://arxiv.org/abs/1706.03762

[6] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting (2021). https://arxiv.org/abs/2012.07436

[7] Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting (2022). https://arxiv.org/abs/2106.13008

[8] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting (2022).

https://arxiv.org/abs/2201.12740

[9] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: iTransformer: Inverted Transformers Are Effective for Time Series Forecasting (2024). https://arxiv.org/abs/2310.06625

[10] Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The Eleventh International Conference on Learning Representations (2023). https://openreview.net/forum?id=vSVLM2j9eie

[11] Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., Long, M.: Timexer: Empowering transformers for time series forecasting with exogenous variables. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024). https://openreview.net/forum?id=INAeUQ04lT

[12] Cheng, X., Chen, X., Li, S., Luo, D., Wang, X., Zhao, D., Yan, R.: Leveraging 2D Information for Long-term Time Series Forecasting with Vanilla Transformers (2024). https://arxiv.org/abs/2405.13810

[13] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A Time Series is Worth 64 Words: Long-term Forecasting with Transformers (2023). https://arxiv.org/abs/2211.14730

[14] Huang, S., Liu, Y., Zhang, F., Li, Y., Li, J., Zhang, C.: Crosswavenet: A dual-channel network with deep cross-decomposition for long-term time series forecasting. Expert Syst. Appl. **238**(Part A), 121642 (2024)

[15] Bai, S., Kolter, J.Z., Koltun, V.: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling (2018). https://arxiv.org/abs/1803.01271

[16] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis (2023). https://arxiv.org/abs/2210.02186

[17] donghao, L., xue: ModernTCN: A modern pure convolution structure for general time series analysis. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=vpJMJerXHU

[18] Jin, M., Koh, H.Y., Wen, Q., Zambon, D., Alippi, C., Webb, G.I., King, I., Pan, S.: A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection (2024). https://arxiv.org/abs/2307.03759

[19] Yan, S., Xiong, Y., Lin, D.: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition (2018). https://arxiv.org/abs/1801.07455

[20] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.: Temporal Graph Networks for Deep Learning on Dynamic Graphs (2020). https://arxiv.org/abs/2006.10637

[21] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=rJXMpikCZ

[22] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are Transformers Effective for Time Series Forecasting? (2022). https://arxiv.org/abs/2205.13504

[23] Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=r1ecqn4YwB

[24] Challu, C., Olivares, K.G., Oreshkin, B.N., Garza, F., Mergenthaler-Canseco, M., Dubrawski, A.: N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting (2022). https://arxiv.org/abs/2201.12886

[25] Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., Kalagnanam, J.: Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. KDD '23, pp. 459–469. ACM, ??? (2023). https://doi.org/10.1145/3580305.3599533 . http://dx.doi.org/10.1145/3580305.3599533

[26] Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J.Y., Zhou, J.: TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting (2024). https://arxiv.org/abs/2405.14616

[27] Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting (2023). https://arxiv.org/abs/2205.14415

[28] Lin, S., Lin, W., Wu, W., Wang, S., Wang, Y.: PETformer: Long-term Time Series Forecasting via Placeholder-enhanced Transformer (2023). https://arxiv.org/abs/2308.04791

[29] Zhang, Y., Ma, L., Pal, S., Zhang, Y., Coates, M.: Multi-resolution Time-Series Transformer for Long-term Forecasting (2024). https://arxiv.org/abs/2311.04147

[30] Shabani, M.A., Abdi, A.H., Meng, L., Sylvain, T.: Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In: The Eleventh International Conference on Learning Representations (2023). https://openreview.net/forum?id=sCrnllCtjoE

[31] Chen, P., ZHANG, Y., Cheng, Y., Shu, Y., Wang, Y., Wen, Q., Yang, B., Guo, C.:

Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=lJkOCMP2aW

[32] Piao, X., Chen, Z., Murayama, T., Matsubara, Y., Sakurai, Y.: Fredformer: Frequency debiased transformer for time series forecasting. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. KDD '24, pp. 2400–2410. ACM, ??? (2024). https://doi.org/10.1145/3637528.3671928 . http://dx.doi.org/10.1145/3637528.3671928

[33] Ni, Z., Yu, H., Liu, S., Li, J., Lin, W.: BasisFormer: Attention-based Time Series Forecasting with Learnable and Interpretable Basis (2024). https://arxiv.org/abs/2310.20496

[34] Yu, C., Wang, F., Shao, Z., Sun, T., Wu, L., Xu, Y.: DSformer: A Double Sampling Transformer for Multivariate Time Series Long-term Prediction (2023). https://arxiv.org/abs/2308.03274

[35] Yang, Y., Zhu, Q., Chen, J.: VCformer: Variable Correlation Transformer with Inherent Lagged Correlation for Multivariate Time Series Forecasting (2024). https://arxiv.org/abs/2405.11470

[36] Liu, J., Liu, C., Woo, G., Wang, Y., Hooi, B., Xiong, C., Sahoo, D.: UniTST: Effectively Modeling Inter-Series and Intra-Series Dependencies for Multivariate Time Series Forecasting (2024). https://arxiv.org/abs/2406.04975

[37] Shu, Y., Lampos, V.: DeformTime: Capturing Variable Dependencies with Deformable Attention for Time Series Forecasting (2025). https://arxiv.org/abs/2406.07438

[38] Żelaszczyk, M., Mańdziuk, J.: Interpretable Multi-task Learning with Shared Variable Embeddings (2024). https://arxiv.org/abs/2405.06330