TIME SERIES FORECASTING MAY BE AS SIMPLE AS CONTROLLING POLYNOMIALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting is commonly formulated as a representation-and-decoding problem, where historical observations are encoded into latent representations and future sequences are subsequently generated from them. In this paradigm, history is primarily treated as a source of information to be encoded and extracted. Modern architectures have made historical representation increasingly adaptive, selective, and expressive. Yet history is still primarily processed to obtain a representation from which the future is subsequently predicted. We ask whether history can play a more direct role: instead of merely being represented, can history actively govern how the future is formed? Based on this perspective, we learn a bank of polynomial predictive functions directly over the future horizon. A lightweight controller converts history into horizon-wise control signals that dynamically modulate their contributions. Complex future dynamics are then synthesized through the history-conditioned coordination of these simple functions. We refer to this model as PolyCtrl, reflecting its history-driven control of polynomial predictive functions. Extensive experiments on eight real-world datasets across multiple forecasting horizons demonstrate that this simple formulation achieves highly competitive forecasting performance. Further analysis reveals a consistent capacity-saturation phenomenon. Once the polynomial order and the size of the predictive-function bank provide sufficient representational capacity, further increasing either of them yields only marginal improvements. Accurate forecasts also rarely depend on only a few dominant functions. Instead, they typically emerge from the coordinated contributions of multiple simple polynomial predictive functions. These findings suggest that the future-generation mechanism required for strong time series forecasting may be considerably simpler than commonly assumed. With a sufficiently rich collection of simple predictive functions and history-driven dynamic control, complex future dynamics can already be effectively modeled.

1 INTRODUCTION

Deep learning models have achieved remarkable progress in time series forecasting, where Transformers, state-space models, MLPs, and specialized temporal architectures have been explored to capture long-term dependencies, multivariate correlations, and non-stationary patterns. Efficient attention mechanisms support long-range forecasting in Informer and Pyraformer (Zhou et al., 2021; Liu et al., 2022b). Decomposition- and frequency-aware Transformers, including Autoformer, FEDformer, and ETSformer, introduce explicit temporal priors into attention-based forecasting (Wu et al., 2021; Zhou et al., 2022b; Woo et al., 2022). Patch- and channel-oriented designs further revise the input representation in PatchTST, iTransformer, and Crossformer (Nie et al., 2023; Liu et al., 2024; Zhang & Yan, 2023). Other specialized temporal representations exploit two-dimensional periodic variation or Legendre and Fourier projections (Wu et al., 2023; Zhou et al., 2022a). Despite these architectural differences, most existing methods follow a similar forecasting paradigm: historical observations are first encoded into predictive representations, which are then transformed or decoded into future values. In this sense, a central question behind existing forecasters has been how to extract richer and more effective predictive information from history.

However, recent studies on simple forecasting models have begun to question whether increasingly complicated architectures are necessary for accurate forecasting. Simple linear mappings can al-

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

ready remain competitive on long-term benchmarks (Zeng et al., 2023), while lightweight MLP forecasters retain nonlinear and covariate modeling capacity (Chen et al., 2023; Das et al., 2023). State-space forecasting likewise offers a linear-complexity alternative to attention-based models (Wang et al., 2025). Considering the effectiveness of these predictors, we reflect on where the complexity of time series forecasting actually needs to reside. Although existing methods continuously improve how historical observations are encoded and represented, history still mainly serves as the information source from which predictive representations are constructed, while the future is subsequently generated from these representations. This motivates us to ask a different question: does history have to remain merely the source of predictive representations, or can it directly determine how the future should be formed?

Concerning the conventional role of history as information to be encoded, we take a different view on the forecasting process. Structured forecast synthesis has shown that explicit bases can support future generation; for example, N-BEATS constructs forecasts through basis expansion (Oreshkin et al., 2020). Instead of relying on a complex prediction network to directly generate future values from latent representations, we explicitly represent the future as a dynamic composition of simple predictive functions. Specifically, we learn a bank of polynomial predictive functions directly defined over the forecasting horizon and employ a lightweight controller to transform historical observations into horizon-wise control signals. These signals dynamically regulate the contribution of each predictive function at different future positions. In this way, the complexity of forecasting no longer needs to be carried by a single complicated future generator, but can emerge from the coordinated composition of multiple simple functions under historical control.

Based on the above motivation, we propose **PolyCtrl** (*History-Driven Control of Polynomial Predictive Functions*) for time series forecasting. The key idea of PolyCtrl is not to construct increasingly expressive future functions, but to reconsider how forecasting complexity should be allocated. The polynomial predictive functions remain structurally simple, while history-dependent variation is mainly handled by the dynamic control mechanism. Therefore, no individual function is required to describe the complete future trajectory. Different functions can contribute differently across future positions and jointly synthesize complex dynamics according to historical observations.

This view further allows us to rethink the source of forecasting capacity. PolyCtrl does not pursue a more sophisticated historical encoder or a more complicated future generator. Instead, it adopts a simple division of labor between history-dependent control and structured future generation. This naturally raises two questions: how many controllable predictive functions are needed to describe complex futures, and how expressive should each individual function be?

To answer these questions, we systematically evaluate PolyCtrl on eight real-world time series datasets under multiple forecasting horizons. Experimentally, this simple formulation achieves highly competitive forecasting performance. More importantly, our analyses reveal that forecasting performance saturates once P and K provide sufficient capacity, and that increasing individual-function complexity cannot substitute for a sufficiently rich function bank. We further find that accurate forecasts are typically formed through the coordinated contributions of multiple polynomial functions rather than a few dominant ones.

Our contributions are threefold:

- We revisit the role of history in time series forecasting and propose a function-control perspective, where historical observations directly govern how explicit future predictive functions are composed.
- We propose PolyCtrl, which uses history-driven, horizon-wise control signals to dynamically coordinate a bank of simple polynomial predictive functions, separating historical control from structured future generation.
- Extensive experiments on eight real-world benchmarks demonstrate competitive forecasting performance. Further analyses of function-bank capacity and contribution patterns show that complex future dynamics emerge from the coordinated composition of multiple simple functions.

2 Related Work

Representation-learning forecasters. Transformer forecasting has developed several complementary mechanisms for long sequences. Informer and Pyraformer reduce the cost of long-range dependency modeling through sparse or pyramidal attention (Zhou et al., 2021; Liu et al., 2022b). Autoformer, FEDformer, and ETSformer combine attention with decomposition, autocorrelation, frequency selection, or exponential-smoothing structure (Wu et al., 2021; Zhou et al., 2022b; Woo et al., 2022). PatchTST, iTransformer, and Crossformer instead emphasize patch-level or channel-level representations for multivariate forecasting (Nie et al., 2023; Liu et al., 2024; Zhang & Yan, 2023). Non-stationary Transformers address distribution shifts, whereas TimeXer incorporates exogenous variables through cross-attention (Liu et al., 2022c; Wang et al., 2024b).

Lightweight and state-space alternatives. The competitiveness of simple linear forecasting is highlighted by DLinear (Zeng et al., 2023). MLP-based or lightweight models explore temporal and feature mixing, dense encoding, sampling-oriented prediction, multiscale mixing, and frequency-domain interpolation (Chen et al., 2023; Das et al., 2023; Zhang et al., 2022; Wang et al., 2024a; Xu et al., 2024; Yi et al., 2023). Selective state-space approaches, including S-Mamba, TimeMachine, and MambaTS, provide another route to long-context modeling with near-linear complexity (Wang et al., 2025; Ahamed & Cheng, 2024; Cai et al., 2024).

Structured future generation. Several approaches introduce explicit temporal structure into forecasting. Decomposition-based methods separately model trend and seasonal components or local and global contexts (Wu et al., 2021; Zhou et al., 2022b; Woo et al., 2022; Wang et al., 2024a; 2023). Frequency-domain approaches use Fourier components, Legendre memory, or joint time–frequency representations (Zhou et al., 2022b; Xu et al., 2024; Yi et al., 2023; Zhou et al., 2022a; Chen et al., 2024). N-BEATS and N-HiTS construct forecasts through basis expansion and hierarchical interpolation, respectively (Oreshkin et al., 2020; Challu et al., 2023); in particular, the interpretable N-BEATS configuration uses explicit polynomial trend bases and Fourier seasonality bases. Other structured temporal models use two-dimensional periodic variation, sample convolution and interaction, or Koopman predictors (Wu et al., 2023; Liu et al., 2022a; 2023).

Different from these representation-learning and forecast-synthesis approaches, PolyCtrl learns global polynomial predictive functions directly over the future horizon and uses history-conditioned, horizon-wise control signals to regulate their contributions. In contrast to N-BEATS, which performs basis expansion through coefficients associated with its forecast basis, PolyCtrl dynamically controls the contribution of each learned predictive function at each future position. Thus, the same predictive function can participate differently across the horizon, leading to a forecasting formulation based on the dynamic coordination of explicit future functions.

3 PolyCtrl

Consider a multivariate historical series $\mathbf{X} \in \mathbb{R}^{L\times C}$, where L denotes the historical lookback length and C is the number of variables, PolyCtrl formulates forecasting as the dynamic composition of explicit future predictive functions. The objective is to predict a future sequence $\mathbf{Y} \in \mathbb{R}^{H\times C}$, where H is the forecasting horizon. PolyCtrl consists of a learnable polynomial function bank and a history-driven controller that determines the contribution of each function.

Normalization. We apply RevIN (Kim et al., 2022) to normalize each channel independently. The normalization statistics are detached from gradient propagation and no learnable affine parameters are used.

3.1 Polynomial Predictive Functions

For each channel, PolyCtrl learns K polynomial predictive functions directly defined over the future horizon. The future position h is mapped to

$$z_h = -1 + \frac{2(h-1)}{H-1}, \qquad h = 1, \ldots, H. \tag{1}$$

The k-th predictive function of channel c is

$$\phi_{c,k}(z_h) = \sum_{p=1}^{P} \theta_{c,k,p} z_h^{p-1}, \qquad k = 1, \ldots, K, \tag{2}$$

where P controls the polynomial complexity. Let $\mathbf{Z} \in \mathbb{R}^{P \times H}$ be the polynomial basis matrix with $Z_{p,h} = z_h^{p-1}$, and $\mathbf{\Theta}_c \in \mathbb{R}^{K \times P}$ be the learnable coefficient matrix for channel c. The predictive-function bank in equation 2 rewritten as

$$\mathbf{\Phi}_c = \mathbf{\Theta}_c \mathbf{Z} \in \mathbb{R}^{K \times H}. \tag{3}$$

Across channels, all coefficients are collected as $\mathbf{\Theta} \in \mathbb{R}^{C \times K \times P}$. Thus, K determines the number of controllable functions, while P determines the complexity of each function.

3.2 History-Driven Function Control

For channel c, let $\mathbf{x}_c = \widetilde{\mathbf{X}}_{:,c}$ denote the normalized historical trajectory from RevIN. A shared controller generates horizon-wise control signals:

$$\mathbf{G}_c = \mathrm{reshape}_{K \times H} \left[\tanh \left(\mathbf{W}_g \, \mathrm{Dropout} \left(\mathrm{GELU}(\mathbf{W}_e \mathbf{x}_c + \mathbf{b}_e) \right) + \mathbf{b}_g \right) \right]. \tag{4}$$

where $\mathbf{W}_e \in \mathbb{R}^{D \times L}$, $\mathbf{W}_g \in \mathbb{R}^{KH \times D}$, and $\mathbf{G}_c \in (-1, 1)^{K \times H}$. The controller parameters are shared across channels, while each channel produces its own control matrix from its historical trajectory.

The controlling signal matrix is defined as

$$\mathbf{C}_c = \mathbf{G}_c \odot \mathbf{\Phi}_c, \tag{5}$$

where each entry $C_{c,k,h}$ represents the contribution of the k-th predictive function at future position h.

The controller regulates the utilization of predictive functions without changing their polynomial parameters. We denote the complete history-driven controller by $f_\psi(\cdot)$, which maps a historical trajectory to the corresponding horizon-wise control matrix:

$$\mathbf{G}_c = f_\psi(\mathbf{x}_c).$$

The normalized prediction is obtained by aggregating all controlled contributions:

$$\widehat{Y}_{h,c}^{\mathrm{norm}} = \sum_{k=1}^{K} C_{c,k,h}. \tag{6}$$

3.3 PolyCtrl Architectures

3.3.1 PolyCtrl

PolyCtrl directly applies the shared controller to the complete normalized history:

$$\mathbf{G}_c = f_\psi(\widetilde{\mathbf{X}}_{:,c}), \tag{7}$$

The forecast is then generated according to Eq. equation 6.

3.3.2 Dual PolyCtrl

Dual PolyCtrl applies two independent PolyCtrl branches to the trend and seasonal components. The normalized sequence is first decomposed as

$$\mathbf{T} = \mathrm{MovingAvg}(\widetilde{\mathbf{X}}), \qquad \mathbf{S} = \widetilde{\mathbf{X}} - \mathbf{T}. \tag{8}$$

The two components are independently processed by PolyCtrl:

$$\mathbf{G}_c^{\mathrm{S}} = f_{\psi^{\mathrm{S}}}(\mathbf{S}_{:,c}), \qquad \mathbf{G}_c^{\mathrm{T}} = f_{\psi^{\mathrm{T}}}(\mathbf{T}_{:,c}), \tag{9}$$

with independent controllers and function banks $\{\psi^{\mathrm{S}}, \mathbf{\Theta}^{\mathrm{S}}\}$ and $\{\psi^{\mathrm{T}}, \mathbf{\Theta}^{\mathrm{T}}\}$.

The final normalized forecast is obtained by combining the outputs of the two branches:

$$\widehat{\mathbf{Y}}^{\,\mathrm{norm}} = \widehat{\mathbf{S}} + \widehat{\mathbf{T}}. \tag{10}$$

Figure 1 summarizes the complete PolyCtrl workflow, from input normalization to history-driven control and forecast synthesis.

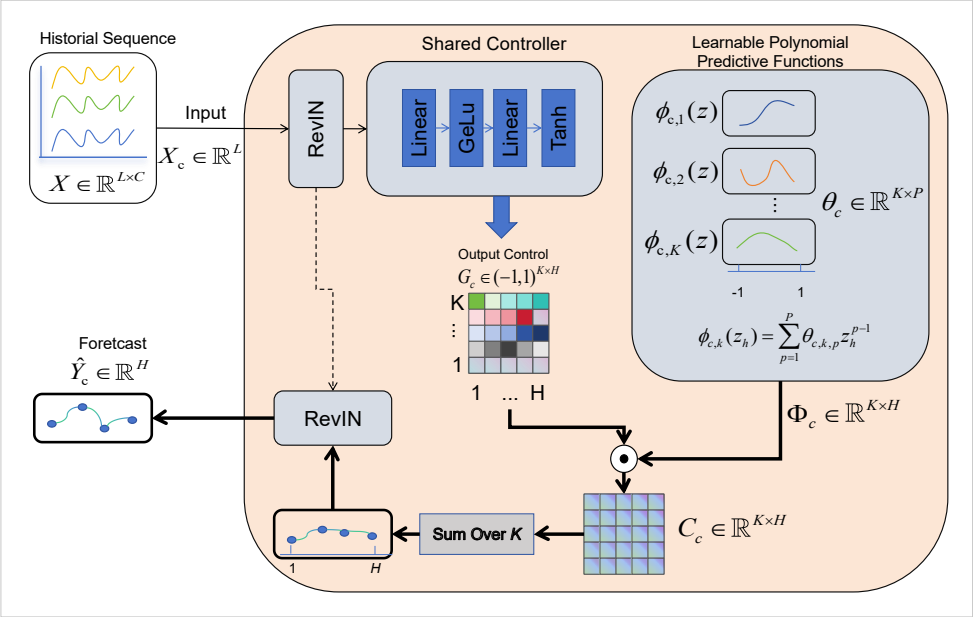


Figure 1: Overall framework of PolyCtrl. A controller shared across channels maps each normalized historical series to control signals $\mathbf{G}_c \in (-1,1)^{K\times H}$, while a channel-specific polynomial bank produces predictive functions $\mathbf{\Phi}_c \in \mathbb{R}^{K\times H}$. Their element-wise interaction forms

4 EXPERIMENTS

4.1 FORECASTING RESULTS

In this section, we conduct extensive experiments to evaluate the forecasting performance of PolyCtrl against representative time series forecasters. We follow the standard long-term forecasting protocol adopted in prior work (Zhou et al., 2021; Wu et al., 2021; Nie et al., 2023) and report results on eight real-world datasets with prediction lengths $H \in \{96, 192, 336, 720\}$, using MSE and MAE as evaluation metrics.

Baselines. We carefully select five representative forecasting methods as our benchmarks, including TimeXer (Wang et al., 2024b), DLinear (Zeng et al., 2023), iTransformer (Liu et al., 2024), CMoS (Si et al., 2025), and SOFTS (Han et al., 2024). These methods cover different forecasting paradigms, ranging from simple linear mappings to modern deep forecasting architectures.For PolyCtrl, we consistently set $K = 25$ and $P = 8$ for all datasets without dataset-specific tuning.

Main results. Table 1 summarizes the forecasting results averaged over four prediction horizons, while the detailed horizon-wise results are reported in Table 5. Overall, Dual PolyCtrl achieves the

best average MSE of 0.314 and the best MSE rank of 2.70, while PolyCtrl obtains the best average MAE of 0.347 and the best MAE rank of 2.56. Notably, the two variants show strong performance on the ETT benchmarks: PolyCtrl performs best on ETTh1, whereas Dual PolyCtrl achieves the best results on ETTh2 and ETTm1 and the lowest MSE on ETTm2. On the remaining datasets, although the best performance varies across different methods, PolyCtrl remains competitive, with Dual PolyCtrl achieving the best MAE on Weather and PolyCtrl the second-best MAE on Solar-H. These results demonstrate the overall effectiveness and robustness of the proposed function-control forecasting framework across diverse multivariate time series.

Table 1: Multivariate forecasting results averaged over $H \in \{96, 192, 336, 720\}$. Lower values are better. Red, blue, and yellow indicate the first-, second-, and third-best results.

Models	**Dual PolyCtrl**		**PolyCtrl**		TimeXer		DLinear		iTransformer		CMoS		SOFTS	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.447	0.439	0.441	0.432	0.466	0.457	0.455	0.451	0.462	0.452	0.452	0.440	0.475	0.461
ETTh2	0.377	0.400	0.381	0.402	0.383	0.408	0.530	0.500	0.388	0.409	0.389	0.406	0.388	0.408
ETTm1	0.381	0.393	0.383	0.394	0.397	0.408	0.403	0.407	0.426	0.420	0.401	0.402	0.405	0.410
ETTm2	0.275	0.320	0.276	0.320	0.286	0.328	0.340	0.390	0.292	0.334	0.283	0.326	0.287	0.330
Weather	0.245	0.272	0.246	0.273	0.244	0.274	0.265	0.316	0.261	0.282	0.253	0.280	0.257	0.278
Electricity	0.188	0.276	0.189	0.277	0.177	0.274	0.212	0.301	0.182	0.272	0.212	0.295	0.184	0.271
Exchange	0.381	0.412	0.378	0.411	0.402	0.422	0.332	0.401	0.380	0.421	0.368	0.408	0.365	0.411
Solar-H	0.218	0.275	0.222	0.262	0.197	0.266	0.252	0.338	0.197	0.259	0.258	0.326	0.198	0.259
Avg.	0.314	0.349	0.315	0.347	0.319	0.355	0.349	0.388	0.323	0.356	0.327	0.360	0.320	0.354
Rank	2.70	2.67	3.11	2.56	3.28	4.19	5.23	5.78	4.81	4.75	4.77	4.06	4.09	3.98

4.2 Parameter Efficiency and Scalability

Parameterization and scalability. For PolyCtrl, the channel-specific polynomial banks contain CKP learnable coefficients. The shared history controller consists of an encoder from L to D dimensions and a projection from D to KH horizon-wise control signals. Including bias terms, the total number of parameters is

$$N_{\text{PolyCtrl}} = CKP + D(L+1) + KH(D+1). \tag{11}$$

Although the polynomial functions could in principle be shared across channels, PolyCtrl assigns an independent function bank to each channel to preserve channel-specific future structures. This introduces only a small parameter overhead, since K and P are both small and almost all model parameters are concentrated in the shared history controller. In particular, each additional channel introduces only KP polynomial coefficients, while the controller parameters remain unchanged with respect to C.

For example, under our unified setting $K = 25$ and $P = 8$, each additional channel requires only 200 parameters. On Electricity ($C = 321$) with $H = 720$, all channel-specific polynomial banks together contain only 64.2K parameters, whereas the shared controller contains approximately 9.28M parameters. Thus, the polynomial banks account for less than 1% of the total parameter count. This allows PolyCtrl to maintain channel-specific predictive functions with negligible parameter overhead.

As shown in Figure 2, PolyCtrl achieves a favorable balance between model size and runtime efficiency on the high-dimensional Electricity dataset. Compared with Dual PolyCtrl, PolyCtrl substantially reduces both FLOPs and parameter count, while nearly doubling the training speed and improving inference throughput by more than 2×. Relative to TimeXer, PolyCtrl requires far fewer FLOPs and achieves markedly faster training and inference. Moreover, although channel-independent DLinear has very low FLOPs, its parameter count increases to 44.84M on Electricity, whereas PolyCtrl requires only 9.35M parameters. Overall, PolyCtrl provides a more balanced efficiency profile by combining compact channel-specific polynomial functions with a controller shared across channels.

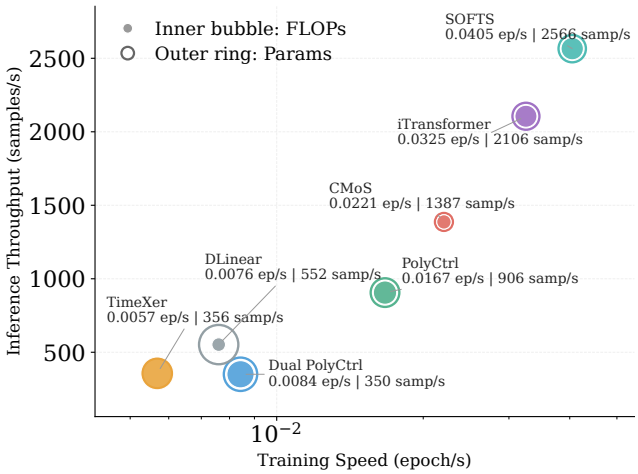


Figure 2: Efficiency comparison on Electricity with prediction length $H = 720$. The horizontal axis denotes training speed (epoch/s), and the vertical axis denotes inference throughput (samples/s). The inner bubble size represents FLOPs, while the outer ring size represents the number of parameters. Models toward the upper-right achieve faster training and inference, while smaller bubbles and rings indicate lower computational and parameter costs.

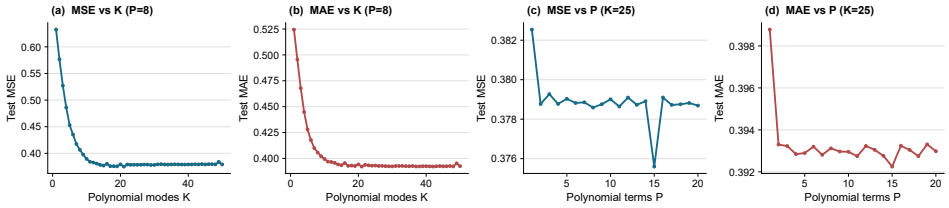


Figure 3: Sensitivity analysis of the function-bank width K and polynomial complexity P on ETTh1 with forecasting horizon $H = 96$. (a)–(b) Test MSE and MAE as K varies with $P = 8$. (c)–(d) Test MSE and MAE as P varies with $K = 25$.

4.3 Sensitivity Analysis of K and P

Figure 3 reveals markedly different roles of K and P. As shown in (a)–(b), increasing K substantially improves both MSE and MAE when the function bank is narrow, followed by clear saturation once a sufficient number of predictive functions is available. In contrast, (c)–(d) show that performance remains relatively stable over a broad range of P after a modest polynomial complexity is reached.

These results indicate that PolyCtrl is considerably more sensitive to the number of independently controllable functions than to the complexity of each individual function. In other words, K primarily provides function diversity, whereas P refines intra-function expressiveness. This observation motivates our unified choice of $K = 25$ and $P = 8$ across all datasets, providing sufficient function-bank width while keeping each predictive function structurally simple.

The distinct empirical effects of K and P are further supported by the expressive-capacity analysis in Appendix A.1–A.5, where we show that P bounds the underlying polynomial function space, while K controls the number of function instances available for independent horizon-wise control.

4.4 Interpretability Analysis

We further investigate how PolyCtrl constructs forecasts in practice. Figure 4 presents a representative example on ETTh1 with forecasting horizon $H = 96$.

Table 2: Dynamic top-m reconstruction statistics across the available forecasting horizons. Entries report the n-weighted mean $\pm$ standard deviation of the number of predictive functions required to recover 90%, 95%, and 99% of the full controlled forecast.

Dataset	$K90$	$K95$	$K99$
Electricity	19.12 ± 3.06	20.79 ± 2.78	23.07 ± 2.28
ETTh1	15.24 ± 1.89	17.55 ± 1.91	20.94 ± 1.89
ETTh2	14.08 ± 1.72	16.29 ± 1.73	19.86 ± 1.87
ETTm1	15.47 ± 2.43	17.31 ± 2.53	20.19 ± 2.82
ETTm2	14.90 ± 2.15	16.90 ± 2.04	20.06 ± 2.14
Exchange	17.41 ± 3.37	18.96 ± 3.41	21.28 ± 3.58
Solar-H	15.08 ± 3.08	16.89 ± 3.46	19.44 ± 4.15
Weather	14.75 ± 4.38	16.46 ± 4.68	18.97 ± 5.26

As shown in Figure 4(a), the learned polynomial modes are individually smooth yet diverse in shape. Their coefficients, visualized in Figure 4(b), are distributed across multiple polynomial terms, indicating that different predictive functions encode different structural tendencies over the future horizon. It is also notable that many modes exhibit relatively larger magnitudes near the two ends of the normalized horizon. This is a natural consequence of polynomial parameterization on $z \in [-1, 1]$: higher-order terms tend to produce larger responses near the boundary locations.

The gate mechanism further reveals how history controls these functions. Figure 4(d) shows that the history-conditioned control signals vary across both modes and future positions, while still exhibiting visible local continuity along the horizon. This behavior is consistent with the nature of time series forecasting, where nearby future steps often evolve smoothly rather than independently. Accordingly, the realized contributions

$$C_{k,h} = g_{k,h}\phi_k(z_h) \tag{12}$$

in Figure 4(c) display structured and horizon-dependent activation patterns. Together, these results show that PolyCtrl does not use fixed predictive functions in a static manner; instead, it dynamically modulates when and how each function contributes to the future trajectory.

Figures 4(e)–(f) further show that the forecast is not dominated by only a few modes. The reconstruction ratio increases gradually as more dynamically selected functions are included, and the top-1 reconstruction alone is clearly insufficient. This sample-level observation is consistent with the dataset-level statistics in Table 2: on average, reconstructing 95% and 99% of the full controlled forecast requires 18.85 and 21.43 out of 25 functions, respectively, and the corresponding values on Electricity further increase to 20.79 and 23.07. This distributed composition differs from frequency-domain approaches that emphasize selected spectral components (Zhou et al., 2022b; Xu et al., 2024); PolyCtrl instead forms its forecasts through the coordinated superposition of many modest modes. Although each basis function is simple and smooth, their history-dependent combination can still generate highly complex future patterns.

5 Conclusion

In this work, we revisit multivariate time series forecasting from a function-control perspective and propose PolyCtrl, where history directly controls how a bank of learnable polynomial predictive functions contributes to the future trajectory. By separating structured future functions from their history-dependent utilization, PolyCtrl provides a simple yet flexible forecasting mechanism, while Dual PolyCtrl further applies this principle independently to seasonal and trend components. Extensive experiments show that PolyCtrl achieves competitive performance and that Dual PolyCtrl obtains the best overall average performance and ranking among the compared methods. Further efficiency and interpretability analyses demonstrate that complex future dynamics can emerge from the coordinated contributions of multiple simple functions under history-dependent control. These results suggest that effective forecasting may rely less on increasingly complicated forecasting operators and more on how simple predictive functions are dynamically composed. As future work, we plan to investigate whether alternative explicit function families, such as sine and cosine functions, exhibit complementary or distinct behaviors compared with polynomial predictive functions under the same control framework.

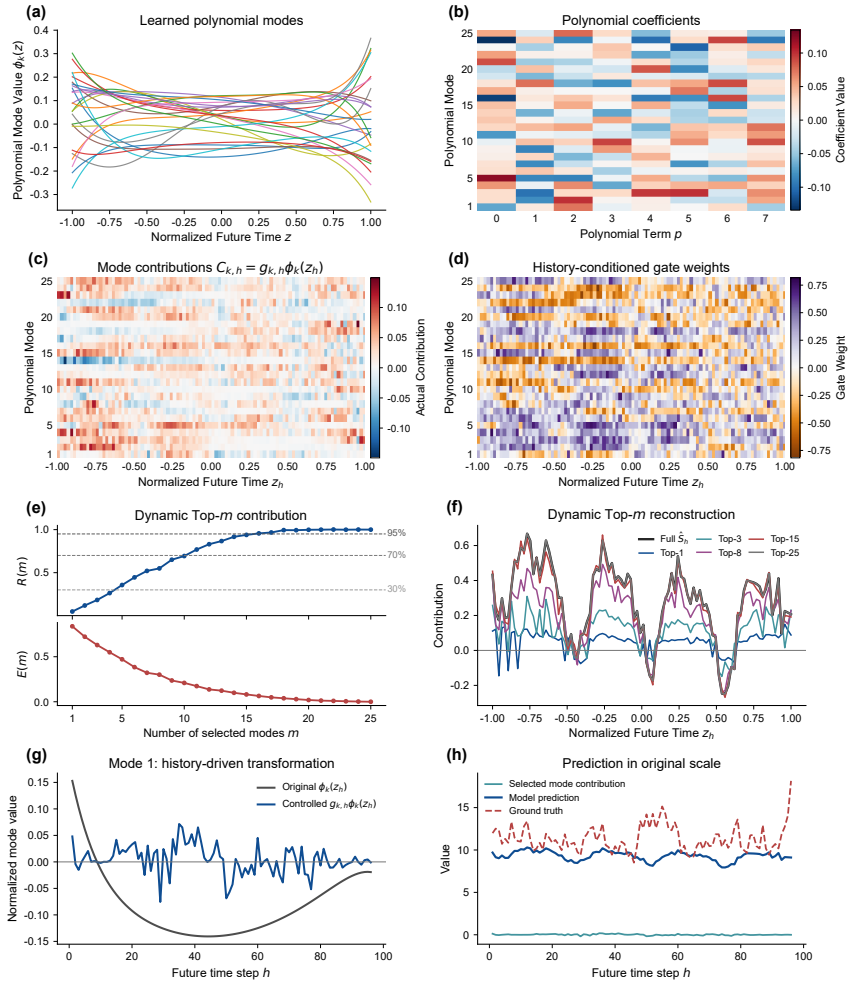


Figure 4: Interpretability analysis of PolyCtrl on ETTh1 with forecasting horizon $H = 96$. (a) Learned polynomial predictive functions. (b) Polynomial coefficients. (c) Realized contributions $C_{k,h} = g_{k,h}\phi_k(z_h)$. (d) History-generated control signals. (e) Dynamic Top-m reconstruction ratio and relative error. (f) Dynamic Top-m reconstruction of the full controlled output. (g) A selected predictive function before and after history-dependent control. (h) Its contribution in the original scale, together with the model prediction and ground truth.

AI USE STATEMENT

In this work, we used generative AI tools for code assistance and language refinement. Specifically, generative AI tools were used to assist with the implementation of model code and visualization scripts, as well as to improve the clarity and readability of the manuscript through translation and language polishing.

The research ideas, model design, theoretical analysis, experimental methodology, and visualization concepts were developed by the authors. Generative AI tools were not used to generate the core research contributions or scientific claims of this work.

All AI-assisted code and text were carefully reviewed, verified, and revised by the authors. We take full responsibility for the final content of this work, including all methods, results, claims, and artifacts presented in the paper.

REFERENCES

Md Atik Ahamed and Qiang Cheng. Timemachine: A time series is worth 4 mambas for long-term forecasting. In *Proceedings of the 27th European Conference on Artificial Intelligence*, 2024.

Xiuding Cai, Yaoyao Zhu, Xueyao Wang, and Yu Yao. Mambats: Improved selective state space models for long-term time series forecasting. *arXiv preprint arXiv:2405.16440*, 2024.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023.

Yushu Chen, Shengzhuo Liu, Jinzhe Yang, Hao Jing, Wenlai Zhao, and Guangwen Yang. A joint time-frequency domain transformer for multivariate time series forecasting. *Neural Networks*, 176:106334, 2024. doi: 10.1016/j.neunet.2024.106334.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K. Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with TiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5816–5828, 2022a.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022b.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022c.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12271–12290, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Haotian Si, Changhua Pei, Jianhui Li, Dan Pei, and Gaogang Xie. CMoS: Rethinking time series prediction through the lens of chunk-wise spatial correlations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 55493–55510, 2025.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024a.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with 10k parameters. In *International Conference on Learning Representations*, 2024.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Defu Lian, Ning An, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures. *arXiv preprint arXiv:2207.01186*, 2022.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved legendre memory model for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12677–12690, 2022a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286, 2022b.

A EXPRESSIVE CAPACITY OF POLYNOMIAL FUNCTION CONTROL

This section analyzes the roles of the number of polynomial terms P, the function-bank width K, and horizon-wise control in PolyCtrl. For clarity, the channel index is omitted when it is not essential.

A.1 POLYNOMIAL CAPACITY AND THE ROLE OF P

Each predictive function in PolyCtrl is parameterized by P polynomial terms:

$$\phi_k(z) = \sum_{p=1}^{P} \theta_{k,p} z^{p-1}. \tag{13}$$

Therefore, $\phi_k \in \Pi_{P-1}$, where

$$\Pi_{P-1} = \mathrm{span}\{1, z, \ldots, z^{P-1}\}. \tag{14}$$

Since the monomials are linearly independent,

$$\dim(\Pi_{P-1}) = P. \tag{15}$$

Thus, P determines the polynomial capacity of each predictive function, with a maximum degree of $P-1$.

Using the matrix formulation in the main text, the predictive-function bank for channel c is

$$\mathbf{\Phi}_c = \mathbf{\Theta}_c \mathbf{Z}, \qquad \mathbf{\Theta}_c \in \mathbb{R}^{K \times P}, \quad \mathbf{Z} \in \mathbb{R}^{P \times H}, \tag{16}$$

where $Z_{p,h} = z_h^{p-1}$. Since the future coordinates $z_1, \ldots, z_H$ are distinct, $\mathbf{Z}$ is a Vandermonde-type evaluation matrix with

$$\mathrm{rank}(\mathbf{Z}) = \min(P, H). \tag{17}$$

It follows that

$$\mathrm{rank}(\mathbf{\Phi}_c) \le \min(K, P, H). \tag{18}$$

When $H \ge P$,

$$\mathrm{rank}(\mathbf{\Phi}_c) \le \min(K, P). \tag{19}$$

Hence, P determines the dimensionality of the polynomial space in which the predictive functions are parameterized.

The polynomial spaces are nested:

$$\Pi_0 \subset \Pi_1 \subset \cdots \subset \Pi_{P-1}. \tag{20}$$

A polynomial of degree $d < P-1$ remains exactly representable in Π_{P-1} by assigning zero coefficients to terms above degree d. The learned coefficients therefore determine the effective polynomial complexity of each predictive function, while P specifies its maximum available capacity.

A.2 HORIZON-WISE FUNCTION COMPOSITION AND THE ROLE OF K

Consider a static combination of K polynomial predictive functions:

$$\widehat{y}(z) = \sum_{k=1}^{K} a_k \phi_k(z), \tag{21}$$

where each coefficient a_k is shared across the entire forecasting horizon. Since $\phi_k \in \Pi_{P-1}$ and the polynomial space is closed under linear combination,

$$\widehat{y}(z) \in \Pi_{P-1}. \tag{22}$$

Static composition therefore uses K global coefficients to combine the predictive functions over the complete future horizon.

PolyCtrl performs horizon-wise composition:

$$\widehat{Y}_{h,c}^{\mathrm{norm}} = \sum_{k=1}^{K} G_{c,k,h}\Phi_{c,k,h}, \tag{23}$$

where the controller produces

$$\mathbf{G}_c = f_\psi(\mathbf{x}_c) \in (-1,1)^{K\times H}. \tag{24}$$

The control coefficient associated with each predictive function can therefore vary across future positions. This changes the composition from K horizon-shared coefficients to a $K \times H$ horizon-dependent control pattern.

For each function k, the controlled trajectory is

$$C_{c,k,h} = G_{c,k,h}\Phi_{c,k,h}, \tag{25}$$

so the same polynomial predictive function can contribute with different magnitudes and signs across the forecasting horizon. The polynomial coefficients determine the shape of each predictive function, while the control matrix determines its history-dependent utilization over future positions.

This horizon-wise composition gives K a role distinct from P. The parameter P determines the polynomial space of each function, whereas K determines the number of predictive function instances available for horizon-wise control. Consequently, $K > P$ provides multiple controllable function instances within the same polynomial space.

This property can be illustrated by considering an extreme case in which all K predictive functions share the same trajectory:

$$\mathbf{\Phi} = \mathbf{1}_K \boldsymbol{\phi}^\top, \qquad \mathrm{rank}(\mathbf{\Phi}) = 1, \tag{26}$$

where $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_H]^\top$. After applying horizon-wise control,

$$\mathbf{C} = \mathbf{G} \odot \mathbf{\Phi} = \mathbf{G}\,\mathrm{Diag}(\boldsymbol{\phi}). \tag{27}$$

If $\phi_h \neq 0$ for all h, then $\mathrm{Diag}(\boldsymbol{\phi})$ is invertible and

$$\mathrm{rank}(\mathbf{C}) = \mathrm{rank}(\mathbf{G}). \tag{28}$$

Thus, even when the predictive-function bank itself has rank one, different horizon-wise control patterns can produce distinct contribution trajectories across the K function instances.

The expressive mechanism of PolyCtrl can therefore be summarized as

$$\underbrace{\mathbf{\Theta}_c\mathbf{Z}}_{\text{predictive functions}} \longrightarrow \underbrace{\mathbf{G}_c \odot \mathbf{\Phi}_c}_{\text{horizon-wise controlled contributions}}. \tag{29}$$

Here, P determines the polynomial capacity of each predictive function, K determines the number of controllable function instances, and $\mathbf{G}_c$ determines their horizon-wise composition conditioned on the historical trajectory.

A.3 Connection to Sensitivity Analysis

The above analysis gives P and K complementary roles:

$$P \;:\; \text{intra-function polynomial capacity}, \qquad K \;:\; \text{number of horizon-wise controllable function instances}. \tag{30}$$

Increasing P enlarges the polynomial space available to each predictive function. Since the polynomial spaces are nested, once sufficient polynomial capacity is available, additional higher-order terms provide progressively more representational capacity within each function.

Increasing K introduces additional predictive function instances that can receive distinct horizon-wise control patterns, providing greater flexibility for dynamic function composition.

These roles are consistent with the sensitivity results in Figure 3. Performance improves substantially when increasing K from a narrow function bank and then gradually saturates, whereas performance remains relatively stable over a broad range of P once a modest polynomial capacity is reached. This behavior supports the use of K primarily for function diversity and controllability and P for intra-function polynomial capacity.

B EXPERIMENTAL DETAILS

B.1 COMPUTATIONAL ENVIRONMENT

All experiments are conducted on a workstation equipped with an AMD Ryzen 5 7500F CPU and an NVIDIA GeForce RTX 4070 SUPER GPU. The GPU power limit is set to 95% throughout the experiments.

B.2 DATASET DESCRIPTION

Table 3 summarizes the datasets used in our experiments. The ETT benchmarks were introduced for long-sequence forecasting in Informer (Zhou et al., 2021). For the remaining datasets, we follow the standard long-term forecasting splits and protocol used in Autoformer, DLinear, and PatchTST (Wu et al., 2021; Zeng et al., 2023; Nie et al., 2023). We evaluate four prediction horizons, i.e., $H \in \{96, 192, 336, 720\}$, for all datasets.

Table 3: Detailed descriptions of the forecasting datasets. *Dim* denotes the number of channels, and *Dataset Size* gives the number of time points in the training, validation, and test sets, respectively.

Dataset	Dim	Dataset Size (Train, Val, Test)	Frequency
ETTh1	7	(8545, 2881, 2881)	Hourly
ETTh2	7	(8545, 2881, 2881)	Hourly
ETTm1	7	(34465, 11521, 11521)	15min
ETTm2	7	(34465, 11521, 11521)	15min
Exchange	8	(5120, 665, 1422)	Daily
Weather	21	(36792, 5271, 10540)	10min
Electricity	321	(18317, 2633, 5261)	Hourly
Solar-H	137	(36601, 5161, 10417)	10min

B.3 IMPLEMENTATION DETAILS

Unless otherwise specified, we use the same training configuration across all datasets to avoid dataset-specific hyperparameter tuning. For PolyCtrl, the number of predictive functions and polynomial terms are fixed to $K = 25$ and $P = 8$, respectively, for all datasets. The detailed experimental configuration is summarized in Table 4.

Table 4: Experimental configuration used across all datasets.

Hyperparameter	Setting
Batch size	32
Learning rate	1×10^{-4}
Optimizer	Adam
Loss function	MSE
Epochs	10
Hidden dimension D	512
Encoder layers	2
Decoder layers	1
FFN dimension	2048
Attention heads	8
Dropout	0.1
Moving average window	25
Number of predictive functions K	25
Polynomial terms P	8

B.4 Complete Forecasting Results

Table 5: Overall forecasting comparison. Red, blue, and yellow denote the first-, second-, and third-best results, respectively.

Dataset	Pred.	PolyCtrl		Dual PolyCtrl		TimeXer		DLinear		iTransformer		CMoS		SOFTS	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.381	0.395	0.375	0.393	0.399	0.412	0.385	0.399	0.390	0.410	0.394	0.402	0.387	0.406
	192	0.432	0.424	0.445	0.436	0.445	0.441	0.440	0.434	0.444	0.436	0.446	0.432	0.437	0.432
	336	0.474	0.445	0.482	0.454	0.466	0.455	0.490	0.469	0.496	0.468	0.487	0.453	0.511	0.477
	720	0.477	0.464	0.485	0.475	0.554	0.519	0.506	0.503	0.518	0.495	0.482	0.472	0.566	0.527
	Avg.	0.441	0.432	0.447	0.439	0.466	0.457	0.455	0.451	0.462	0.452	0.452	0.440	0.475	0.461
ETTh2	96	0.289	0.339	0.284	0.337	0.296	0.348	0.336	0.389	0.294	0.344	0.297	0.344	0.293	0.345
	192	0.368	0.389	0.366	0.388	0.384	0.403	0.448	0.457	0.384	0.403	0.383	0.396	0.386	0.403
	336	0.427	0.432	0.421	0.429	0.427	0.439	0.554	0.518	0.429	0.435	0.430	0.434	0.441	0.441
	720	0.441	0.448	0.437	0.445	0.423	0.442	0.783	0.637	0.444	0.455	0.446	0.449	0.430	0.443
	Avg.	0.381	0.402	0.377	0.400	0.383	0.408	0.530	0.500	0.388	0.409	0.389	0.406	0.388	0.408
ETTm1	96	0.326	0.362	0.323	0.359	0.333	0.366	0.344	0.371	0.342	0.376	0.340	0.370	0.328	0.364
	192	0.358	0.377	0.356	0.377	0.375	0.394	0.382	0.392	0.398	0.404	0.380	0.388	0.380	0.395
	336	0.391	0.400	0.390	0.400	0.413	0.418	0.413	0.413	0.458	0.435	0.410	0.407	0.427	0.424
	720	0.457	0.437	0.457	0.438	0.469	0.453	0.473	0.451	0.506	0.465	0.475	0.441	0.483	0.456
	Avg.	0.383	0.394	0.381	0.393	0.397	0.408	0.403	0.407	0.426	0.420	0.401	0.402	0.405	0.410
ETTm2	96	0.174	0.255	0.175	0.256	0.181	0.262	0.191	0.289	0.188	0.268	0.178	0.259	0.180	0.263
	192	0.239	0.298	0.240	0.299	0.247	0.304	0.275	0.352	0.251	0.312	0.245	0.305	0.244	0.304
	336	0.297	0.335	0.294	0.334	0.309	0.345	0.372	0.419	0.313	0.350	0.307	0.343	0.312	0.349
	720	0.395	0.394	0.391	0.393	0.405	0.400	0.521	0.501	0.414	0.406	0.404	0.398	0.410	0.404
	Avg.	0.276	0.320	0.275	0.320	0.286	0.328	0.340	0.390	0.292	0.334	0.283	0.326	0.287	0.330
Weather	96	0.164	0.208	0.162	0.207	0.156	0.203	0.196	0.256	0.179	0.220	0.172	0.219	0.172	0.212
	192	0.210	0.250	0.208	0.248	0.208	0.252	0.236	0.295	0.223	0.255	0.218	0.258	0.223	0.256
	336	0.266	0.291	0.264	0.290	0.266	0.295	0.281	0.331	0.283	0.301	0.274	0.297	0.277	0.296
	720	0.346	0.343	0.345	0.343	0.345	0.346	0.345	0.382	0.359	0.351	0.350	0.345	0.355	0.348
	Avg.	0.246	0.273	0.245	0.272	0.244	0.274	0.265	0.316	0.261	0.282	0.253	0.280	0.257	0.278
Electricity	96	0.164	0.255	0.163	0.254	0.144	0.242	0.197	0.281	0.154	0.245	0.187	0.273	0.155	0.245
	192	0.173	0.263	0.172	0.262	0.165	0.264	0.196	0.285	0.166	0.256	0.197	0.281	0.169	0.257
	336	0.190	0.279	0.189	0.278	0.182	0.279	0.210	0.302	0.185	0.276	0.211	0.296	0.186	0.274
	720	0.229	0.312	0.227	0.311	0.218	0.311	0.245	0.334	0.222	0.309	0.255	0.329	0.226	0.310
	Avg.	0.189	0.277	0.188	0.276	0.177	0.274	0.212	0.301	0.182	0.272	0.212	0.295	0.184	0.271
Exchange	96	0.089	0.207	0.089	0.206	0.087	0.205	0.082	0.206	0.096	0.218	0.086	0.206	0.086	0.206
	192	0.186	0.304	0.186	0.304	0.191	0.309	0.168	0.304	0.206	0.326	0.180	0.303	0.187	0.309
	336	0.344	0.423	0.346	0.424	0.379	0.446	0.307	0.419	0.380	0.447	0.337	0.419	0.361	0.440
	720	0.893	0.711	0.904	0.715	0.953	0.728	0.771	0.675	0.839	0.693	0.871	0.703	0.825	0.691
	Avg.	0.378	0.411	0.381	0.412	0.402	0.422	0.332	0.401	0.380	0.421	0.368	0.408	0.365	0.411
Solar-H	96	0.227	0.263	0.223	0.273	0.201	0.273	0.254	0.334	0.203	0.258	0.266	0.328	0.203	0.256
	192	0.225	0.263	0.220	0.275	0.200	0.269	0.256	0.338	0.197	0.258	0.264	0.329	0.198	0.259
	336	0.215	0.257	0.212	0.271	0.192	0.260	0.246	0.335	0.194	0.259	0.249	0.321	0.194	0.259
	720	0.221	0.266	0.218	0.282	0.193	0.262	0.251	0.346	0.195	0.263	0.254	0.325	0.198	0.262
	Avg.	0.222	0.262	0.218	0.275	0.197	0.266	0.252	0.338	0.197	0.259	0.258	0.326	0.198	0.259
Overall Avg.		0.315	0.347	0.314	0.349	0.319	0.355	0.349	0.388	0.323	0.356	0.327	0.360	0.320	0.354
Avg. Rank		3.11	2.56	2.70	2.67	3.28	4.19	5.23	5.78	4.81	4.75	4.77	4.06	4.09	3.98

C Additional Interpretability Analyses

C.1 All-Channel Forecasting Waveforms

Figure 5 visualizes the $H = 96$ forecasting window for sample 0 of ETTh1 across all seven channels. The observed historical series are shown in gray, and the dashed vertical line marks the boundary between history and the prediction horizon.

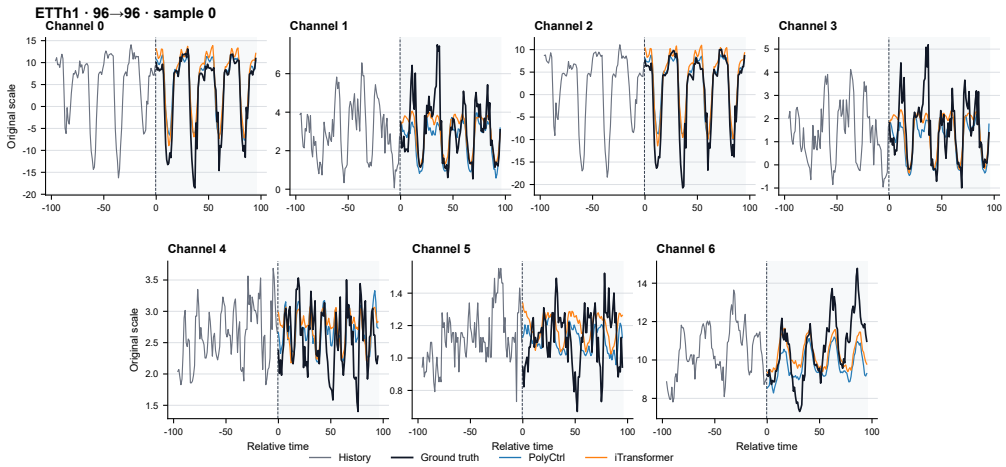


Figure 5: All-channel forecasting waveforms on ETTh1 for $H = 96$ and sample 0. Ground truth is shown in black; PolyCtrl and iTransformer forecasts are shown in blue and orange, respectively.

C.2 Batch-Group Interpretability Panels

This section presents batch-group interpretability views across datasets at forecasting horizon $H = 96$ for sample 0 and channel 0 (Figures 6–13). Each eight-panel figure summarizes learned polynomial modes, polynomial coefficients, realized mode contributions, history-conditioned gate weights, dynamic Top-m reconstruction statistics, controlled-output reconstruction, history-dependent mode control, and original-scale predictions.

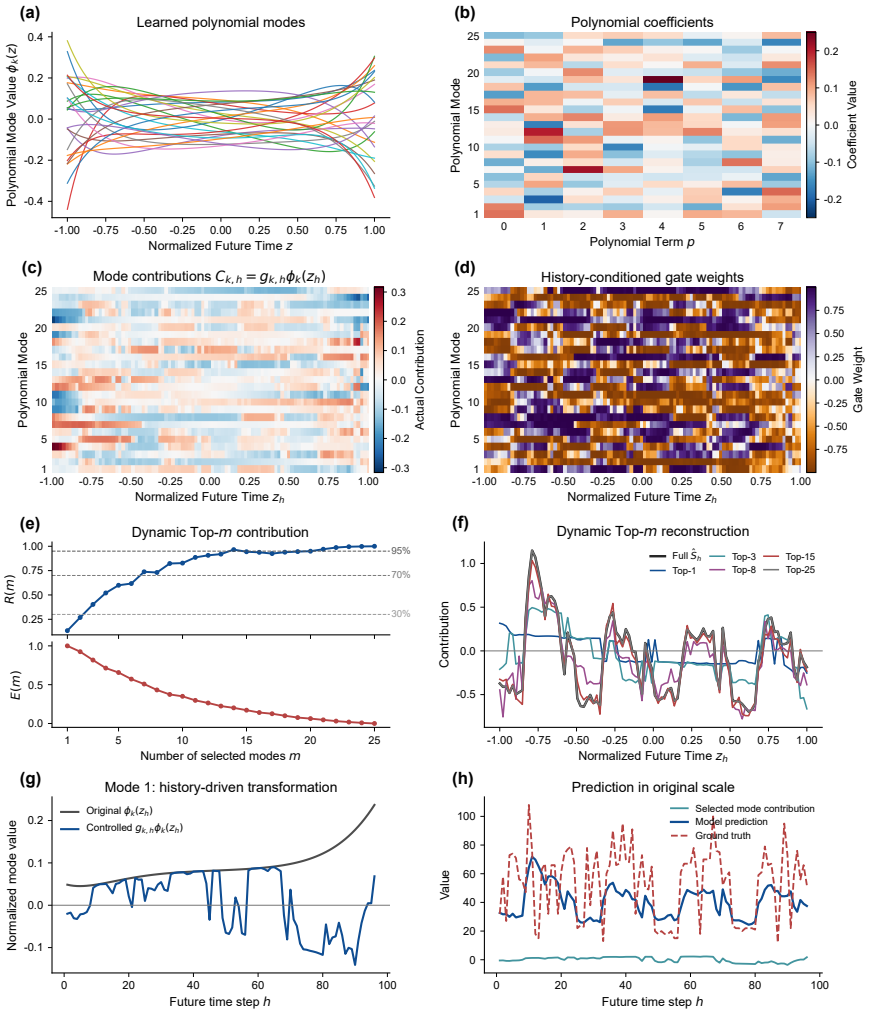


Figure 6: Batch-group interpretability analysis for Electricity ($H = 96$, sample 0, channel 0).

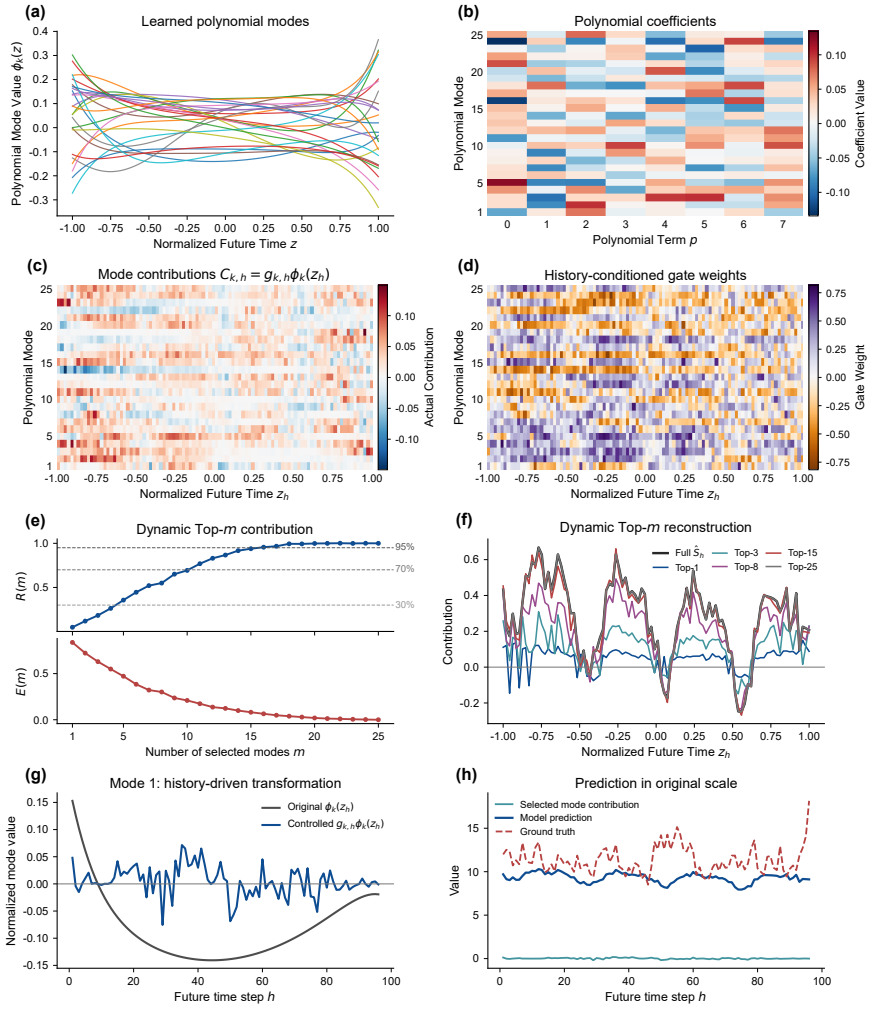


Figure 7: Batch-group interpretability analysis for ETTh1 ($H = 96$, sample 0, channel 0).

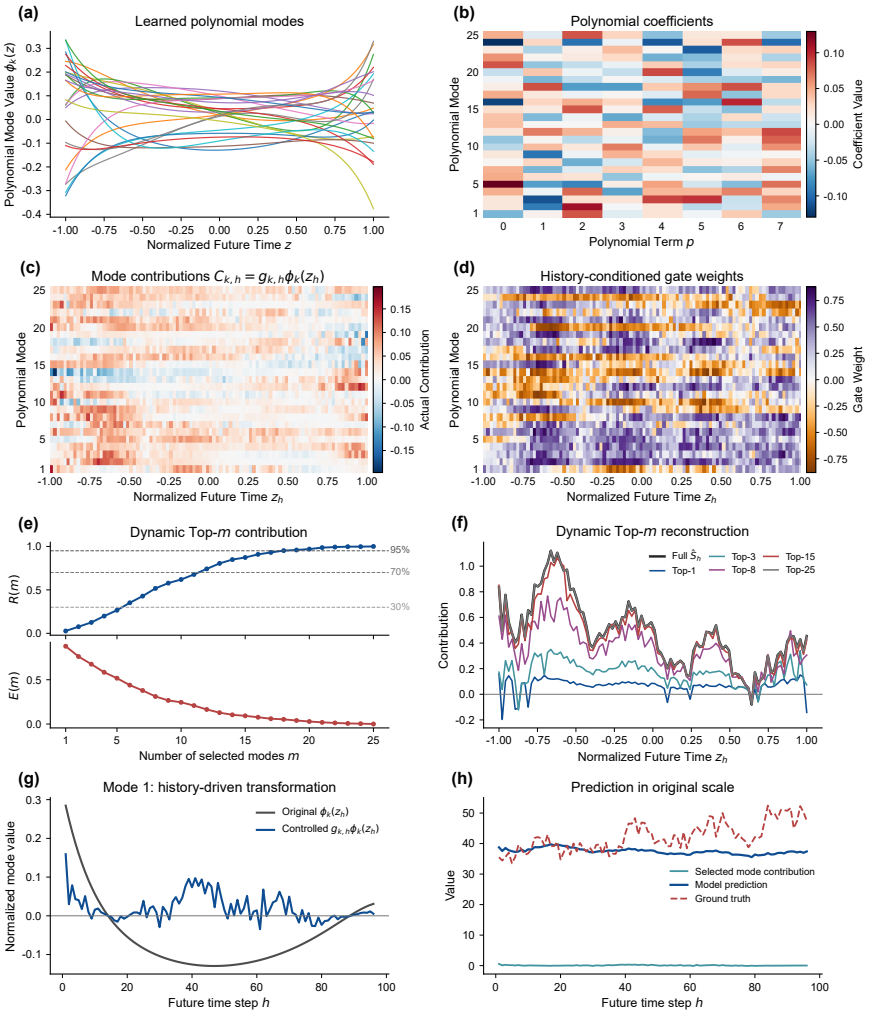


Figure 8: Batch-group interpretability analysis for ETTh2 ($H = 96$, sample 0, channel 0).

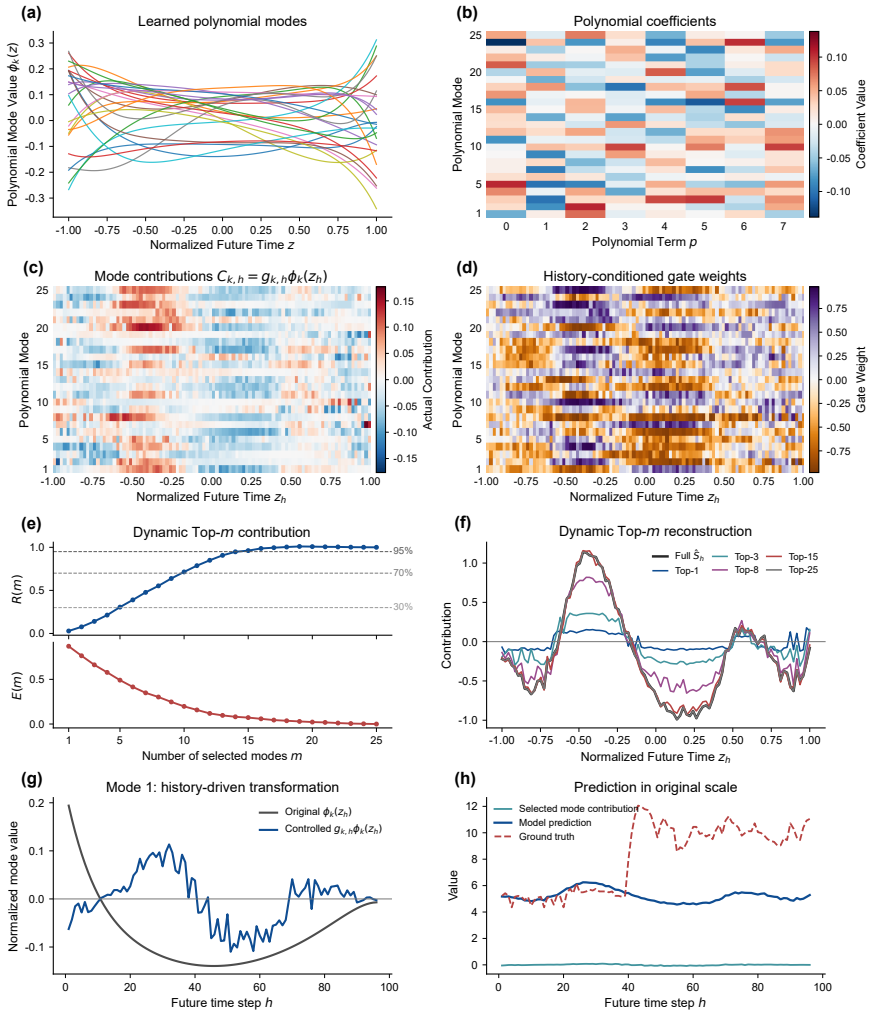


Figure 9: Batch-group interpretability analysis for ETTm1 ($H = 96$, sample 0, channel 0).

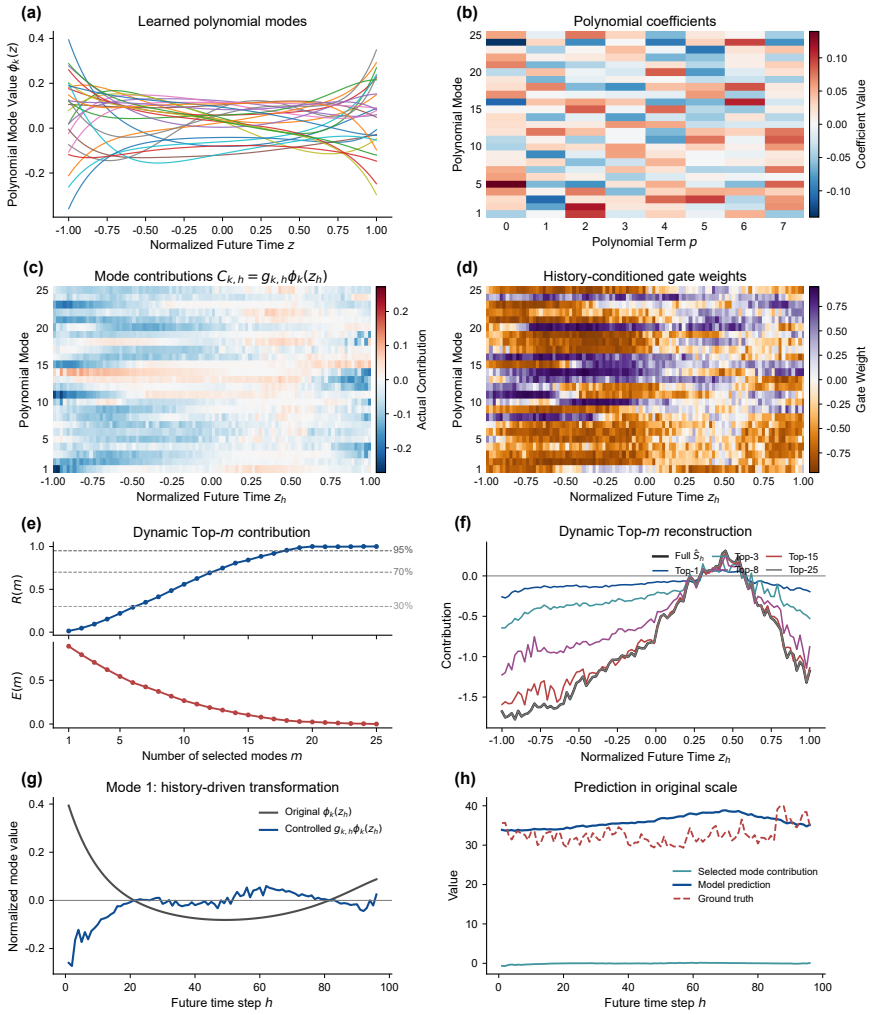


Figure 10: Batch-group interpretability analysis for ETTm2 ($H = 96$, sample 0, channel 0).

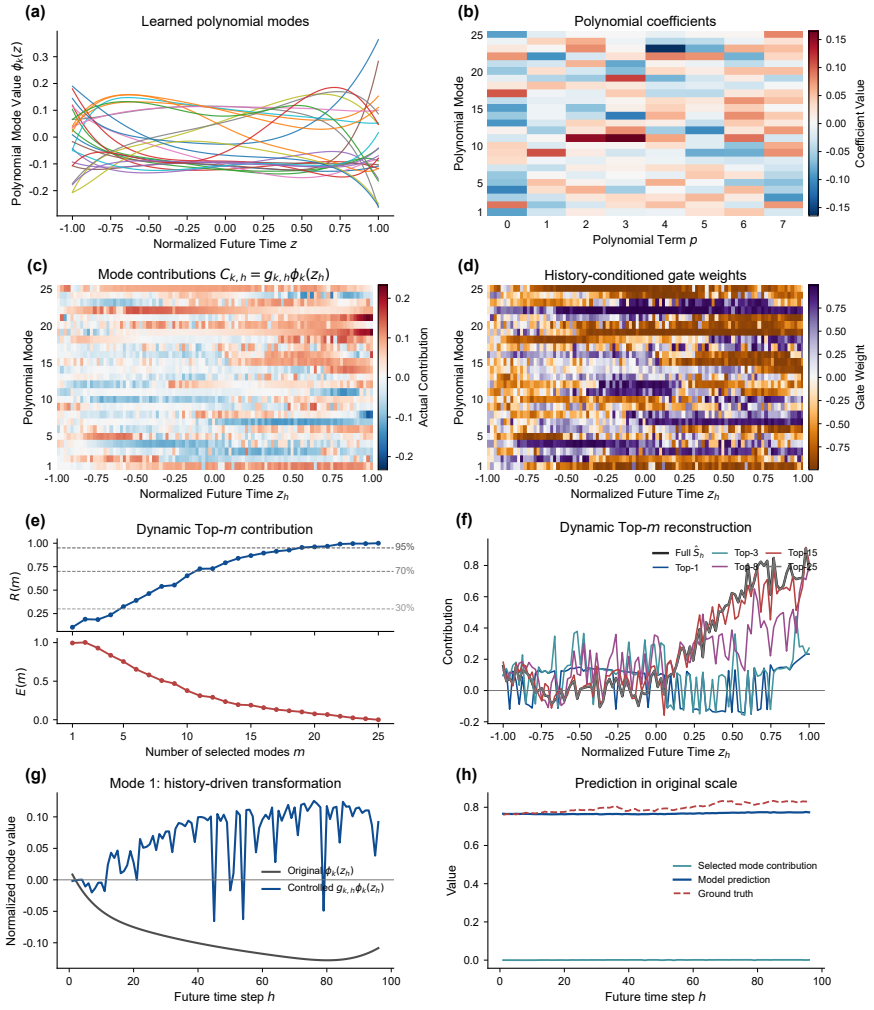


Figure 11: Batch-group interpretability analysis for Exchange ($H = 96$, sample 0, channel 0).

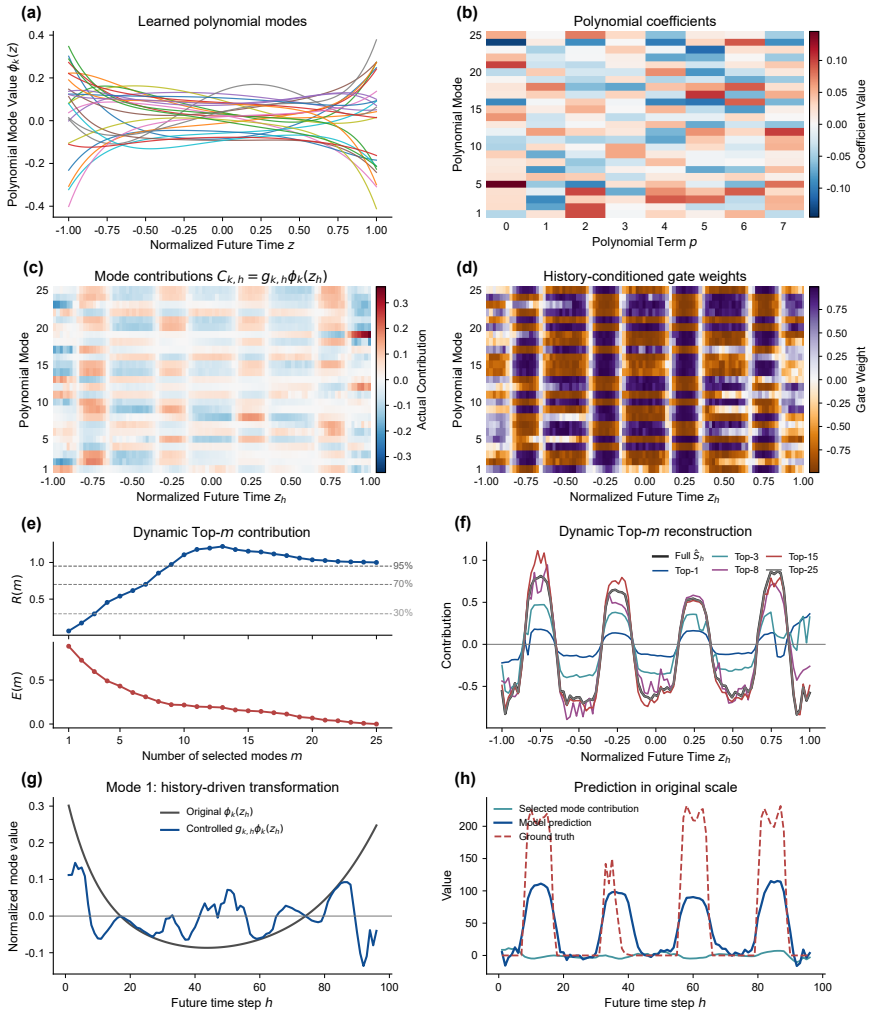


Figure 12: Batch-group interpretability analysis for Solar-H ($H = 96$, sample 0, channel 0).

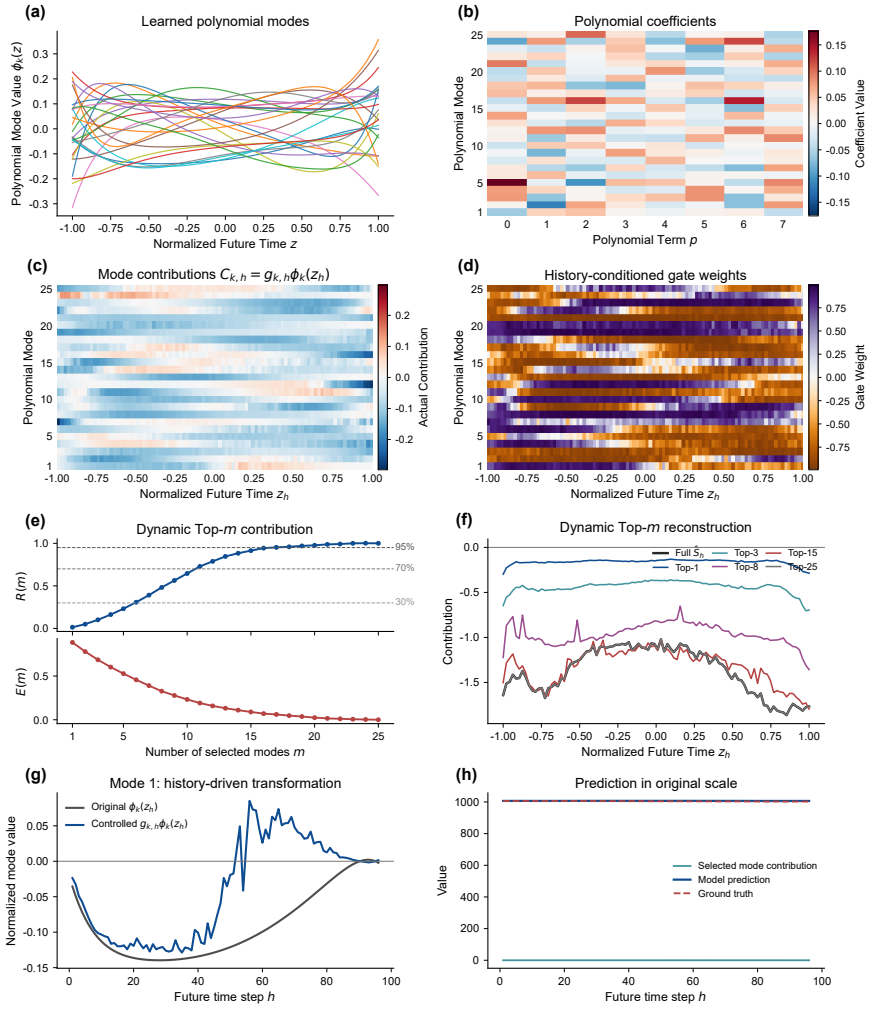


Figure 13: Batch-group interpretability analysis for Weather ($H = 96$, sample 0, channel 0).