HYPERDIFF: HIGH-ORDER STRUCTURE-DRIVEN DIFFUSION FOR MULTIVARIATE PROBABILISTIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate probabilistic forecasting must model both future uncertainty and how that uncertainty is organized across interacting variables. Existing generative forecasters can represent flexible predictive distributions, but they rarely use dynamic group-wise dependence as an explicit uncertainty prior. We propose HyperDiff, a high-order structure-driven diffusion framework that treats a window-conditioned hypergraph as a data-dependent structural prior. HyperDiff learns group-wise relations from each historical window, uses them to organize history-derived volatility while preserving variable-specific heteroscedasticity, and feeds the resulting conditions into a variable-specific diffusion schedule. A deterministic history-based forecast provides the trend center, allowing reverse diffusion to focus on stochastic deviations around a plausible trajectory. Experiments on seven multivariate benchmarks and four forecasting horizons show that HyperDiff achieves lower CRPS and QICE than the evaluated probabilistic forecasting baselines. These results support high-order structural conditioning for multivariate probabilistic forecasting. Code is available at `https://anonymous.4open.science/r/HyperDiff-1D2B`.

1 INTRODUCTION

Multivariate probabilistic time-series forecasting aims to model the conditional distribution of future observations given historical context, supporting risk-aware decision making in applications such as energy, transportation, and finance. This problem is challenging because predictive uncertainty can vary substantially across variables and future positions, while cross-variable dependencies may change with the current system state. Reliable forecasts therefore need to capture both variable-specific uncertainty and its association with the current multivariate organization Gneiting & Katzfuss (2014). Existing probabilistic forecasters typically characterize local uncertainty through parameterized distributions Salinas et al. (2020), while recent work has also begun to consider cross-variable dependence in predictive uncertainty modeling Zheng & Sun (2024).

Recent probabilistic forecasting has increasingly adopted flexible generative frameworks such as normalizing flows and diffusion models Rasul et al. (2021b); Yan et al. (2024). These approaches improve the representation of non-Gaussian and heteroscedastic predictive distributions, and recent methods further adapt generation using nonstationary characteristics or dynamic relational information Ye et al. (2025); Zhao et al. (2024). In parallel, multivariate dependency modeling has progressed toward dynamic and high-order structures that capture group-wise interactions beyond pairwise relations. Dynamic hypergraph methods model changing group organization across temporal states Yin et al. (2024), while adaptive multiscale hypergraphs further extend this capability Shang et al. (2024).

However, these two lines of research still assign different roles to structural information. Probabilistic forecasters mainly characterize uncertainty through local statistics, volatility, or generic relational signals, whereas high-order structures are primarily used for predictive representation and dependency modeling. As a result, the dynamic group-wise organization in the current historical window remains underexplored as a mechanism for organizing predictive uncertainty. This moti-

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.

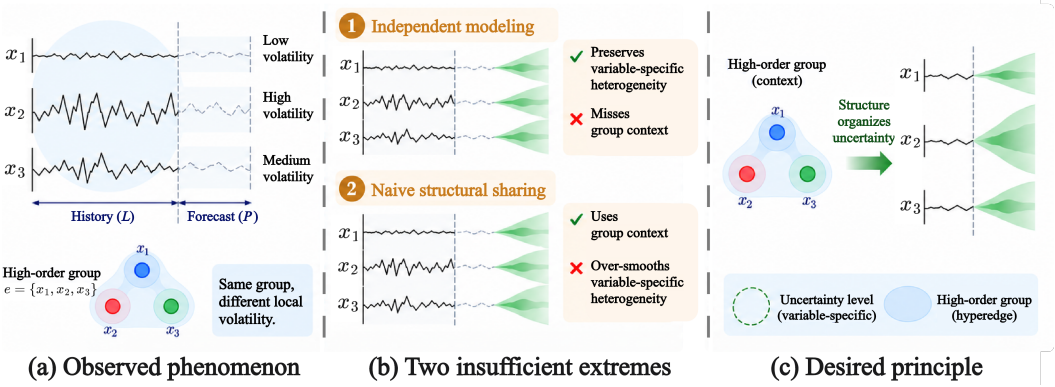


Figure 1: Motivation for high-order uncertainty organization. (a) Variables in the same high-order group can exhibit different local volatility. (b) Independent modeling and naive structural sharing represent two insufficient extremes. (c) High-order structure should organize uncertainty while preserving variable-specific heterogeneity.

vates the central question of this work: *How can window-dependent high-order organization be incorporated into predictive uncertainty modeling and probabilistic generation?*

Directly sharing uncertainty according to high-order structure is insufficient. As illustrated in Figure 1, variables in the same structural group may share useful group-level context while retaining substantially different local volatility. Independent uncertainty modeling preserves such heterogeneity but ignores group context, whereas naive structural sharing may over-smooth variable-specific differences. We therefore argue that high-order structure should *organize* predictive uncertainty rather than homogenize it. Because the relevant group-wise relations also vary across historical windows, the structural information should be learned dynamically and treated as a window-conditioned, data-dependent prior.

To this end, we propose **HyperDiff**, a high-order structure-driven diffusion framework for multivariate probabilistic forecasting. HyperDiff learns a window-conditioned high-order structural prior and uses it to organize history-derived volatility while preserving variable-specific heteroscedasticity. The resulting structural and volatility conditions jointly adapt stochasticity during diffusion, extending high-order dependency modeling from predictive representation to predictive uncertainty organization and probabilistic generation.

The main contributions of this work are summarized as follows:

- **Dynamic high-order variance modeling.** We use a window-conditioned hypergraph to regularize history-derived variance while preserving variable-specific heteroscedasticity.
- **Structure-adaptive heteroscedastic diffusion.** The refined high-order structure and variance field jointly adapt variable-specific diffusion schedules around a deterministic forecast center.
- **Comprehensive empirical evaluation.** Experiments on seven benchmarks and four horizons demonstrate consistent gains in probabilistic accuracy and calibration, with additional analyses of joint-distribution quality, robustness, and efficiency.

2 Problem Formulation

We consider multivariate probabilistic forecasting. Given a historical window $X_{1:L} \in \mathbb{R}^{L\times N}$ with N variables, the goal is to model the conditional distribution of the future window

$$Y = X_{L+1:L+P} \in \mathbb{R}^{P\times N} \tag{1}$$

namely $p_\theta(Y \mid X_{1:L})$, rather than only producing a point forecast.

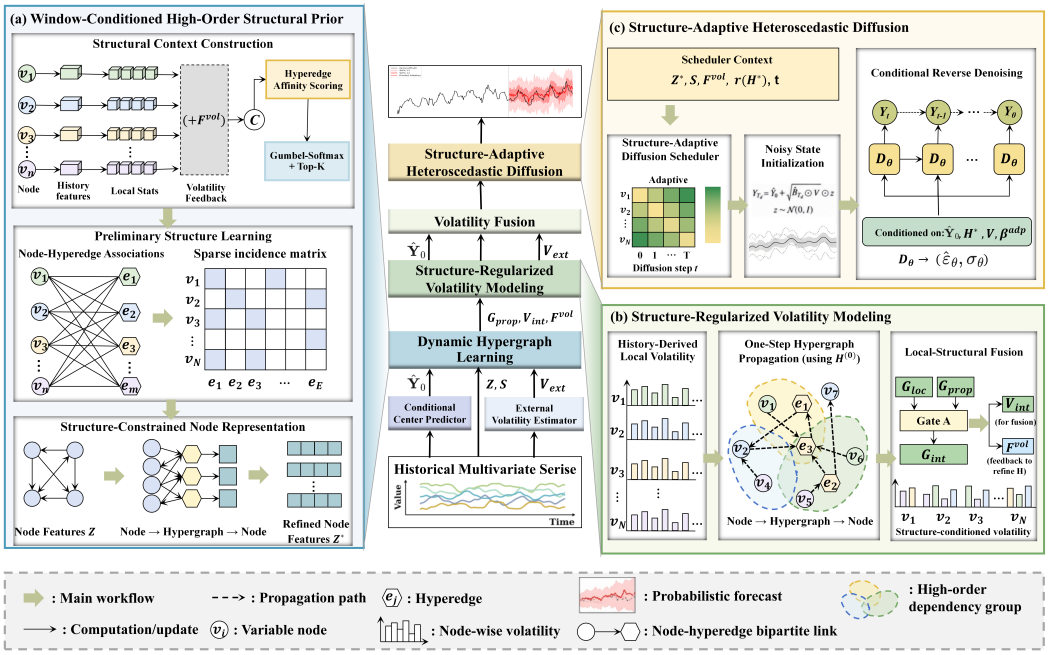


Figure 2: Overall framework of HyperDiff. (a) Window-conditioned high-order structure learning. (b) Structure-regularized volatility modeling with one-step feedback. (c) Structure-adaptive heteroscedastic diffusion forecasting.

In multivariate systems, predictive uncertainty is generally heteroscedastic across variables and future positions. The associated cross-variable organization may also change across historical windows. We therefore consider a window-conditioned high-order structure

$$H(X_{1:L}) \in \mathbb{R}_{+}^{N \times E} \tag{2}$$

as a data-dependent structural prior describing the current group-wise organization among variables. Importantly, $H(X_{1:L})$ is not assumed to represent a ground-truth uncertainty graph or a causal dependency structure.

The problem is how to use this dynamic high-order organization to share history-derived volatility evidence without suppressing variable-specific heteroscedasticity. We then incorporate the resulting structural and volatility conditions into the stochastic generation process for learning $p_\theta(Y \mid X_{1:L})$.

3 Methodology

Multivariate predictive uncertainty depends on both variable-specific, time-varying volatility and the cross-variable organization of the current system state. HyperDiff does not treat the learned high-order structure as a direct estimate of future uncertainty. Instead, it uses the structure inferred from each historical window as a data-dependent prior that organizes local volatility evidence while preserving variable-specific heteroscedasticity. HyperDiff first infers a preliminary structure and uses it to condition history-derived volatility. This volatility state then provides evidence for one structural refinement. The final variance field determines the heteroscedastic prediction scale, while the refined structure adapts variable-specific diffusion scheduling and reverse denoising. Figure 2 illustrates the overall framework.

3.1 Window-Conditioned High-Order Structural Prior

Cross-variable relations need not remain invariant across historical windows. The same variables may participate in different group-wise interactions under different system states, making a globally

shared structure less suited to representing the organization relevant to the current forecast. HyperDiff therefore constructs a high-order structural prior separately for each historical window.

For variable i, the history encoder produces a temporal representation z_i. We additionally summarize its observed local state as

$$s_i = [\mu_i, \sigma_i, \delta_i, \rho_i] \tag{3}$$

where the components denote the historical mean, standard deviation, average first-order change, and maximum absolute first-order change, respectively. These descriptors characterize the current historical state rather than directly estimating future uncertainty. Let $\widetilde{s}_i = \phi_s(s_i)$. For notational convenience, c_i denotes the context used in one structural encoding pass, combining the node representation, the projected local state, and optional volatility feedback. The feedback component is set to zero during the preliminary pass and replaced by the volatility feature introduced in Section 3.2 during structural refinement.

The affinity between variable i and candidate hyperedge e is defined as

$$a_{ie} = [\phi_a(c_i)]_e + \kappa \, \mathrm{sim}(c_i, q_e) + [\phi_g(\widetilde{s}_i)]_e \tag{4}$$

where q_e is a learnable hyperedge prototype. The first term makes the assignment responsive to the current historical representation. The prototype term provides a persistent reference for candidate groups across windows. The local-state term exposes changes in level, scale, and recent dynamics that may not be fully preserved in the latent representation.

We convert the affinities into differentiable memberships with Gumbel–Softmax and retain the Top-K candidate hyperedges for each variable. This produces a sparse, window-conditioned incidence matrix $H \in \mathbb{R}_+^{N \times E}$. The structural encoder subsequently forms group-level context through structure-constrained variable interaction and node–hyperedge–node aggregation. Details of sparse incidence construction and structural encoding are provided in Appendix C.1.

Before structure-aware volatility is available, we denote the preliminary structure by $H^{(0)} = f_H(X_{1:L}; 0)$. This structure identifies variables that share group-level context under the current historical state, but it does not prescribe their predictive uncertainty. Variables in the same high-order relation may still have different volatility levels. We therefore use $H^{(0)}$ to organize volatility evidence, not as an uncertainty graph.

3.2 Structure-Regularized Volatility Modeling and Refinement

Structural relatedness does not imply identical predictive uncertainty. HyperDiff therefore uses high-order structure to regularize, rather than replace, history-derived local volatility evidence.

A history-only volatility encoder first constructs the horizon-aligned representation $G_{\mathrm{loc}} = f_{\mathrm{loc}}(X_{1:L}) \in \mathbb{R}^{N \times P}$. Because it is derived from each variable's observed history, G_{loc} preserves local variation before structural conditioning. Let $L_{H^{(0)}}$ denote the normalized hypergraph Laplacian induced by the preliminary structure. We define the desired structure-conditioned representation as

$$U^\star = \arg\min_U \left\{ \frac{1}{2} \|U - G_{\mathrm{loc}}\|_F^2 + \frac{\lambda}{2} \, \mathrm{tr}\left(U^\top L_{H^{(0)}} U\right) \right\} \tag{5}$$

Equation (5) balances local fidelity and structural consistency. The first term anchors the representation to history-derived evidence, whereas the second penalizes deviations that are inconsistent with the current high-order organization. Importantly, the regularizer neither assigns uncertainty to a hyperedge nor requires its members to have equal variance.

We do not solve Eq. (5) explicitly during training. Appendix B.1 shows that its solution admits a fixed-point form in which structural information acts as a correction to the local representation. HyperDiff realizes this idea with one learnable node–hyperedge–node correction,

$$G_{\mathrm{prop}} = \mathcal{P}_{H^{(0)}}(G_{\mathrm{loc}}) \tag{6}$$

To make the correction strength sample-, variable-, and forecast-position-dependent, we define

$$G_{\mathrm{int}} = A \odot G_{\mathrm{loc}} + (1 - A) \odot G_{\mathrm{prop}} \tag{7}$$

Here, A adaptively interpolates between the local and structurally propagated representations, generalizing the fixed trade-off in Eq. (5). Its relation to the one-step fixed-point correction is detailed in Appendix B.2.

The integrated representation produces a positive structure-conditioned variance estimate V_{int} and a latent volatility-feedback feature F^{vol}. Because $H^{(0)}$ is inferred before structure-conditioned volatility is available, F^{vol} provides additional evidence for one structural refinement:

$$H^{\star} = f_H\left(X_{1:L}; F^{\mathrm{vol}}\right) \tag{8}$$

This is a one-step feedback rather than an iterative co-optimization: $H^{(0)}$ conditions the variance representation, and the resulting volatility state refines the structure once. The refined $H^{\star}$ is not fed back to recompute V_{int}.

To retain a complementary history-only variance estimate outside the hypergraph branch, HyperDiff also computes V_{ext} from temporal variation. The final history-conditioned variance field is

$$V = \gamma \odot V_{\mathrm{ext}} + (1-\gamma) \odot V_{\mathrm{int}} \tag{9}$$

The two gates operate at different stages: A controls the strength of the structural correction, whereas γ balances the structure-conditioned and direct temporal variance estimates.

The resulting V provides the history-conditioned variance field over future positions and variables, while $H^{\star}$ represents the refined high-order organization. The next section uses both quantities to adapt stochastic generation.

3.3 Structure-Adaptive Heteroscedastic Diffusion

The previous stage provides a history-conditioned variance field and a refined high-order state. Prediction-space heteroscedasticity, however, is distinct from the noise injected at each diffusion step. HyperDiff therefore assigns different roles to $V_{b,p,i}$ and $\beta^{\mathrm{adp}}_{b,t,i}$: the former represents the history-conditioned predictive variance of variable i at future position p, whereas the latter controls its perturbation strength at diffusion step t.

To make the diffusion trajectory responsive to the current multivariate state, we summarize each variable's structural role from $H^{\star}$ and combine it with the refined node representation, local descriptors, volatility feedback, and diffusion-step embedding to form the scheduling context $u_{b,t,i}$. The exact structural statistics and scheduler context are specified in Appendix C.1. The adaptive schedule is

$$\beta^{\mathrm{adp}}_{b,t,i} = \mathrm{Clip}_{[10^{-6},\,0.999]}\left(\beta_t\left[0.5 + 1.5\ \mathrm{Sigmoid}\left(\mathcal{M}(u_{b,t,i}) + \zeta_i\right)\right]\right) \tag{10}$$

where β_t is the base diffusion schedule and ζ_i is a learnable variable-specific scheduling bias.

The global schedule preserves the overall diffusion trajectory, while the learned multiplicative factor redistributes perturbation strength across variables and windows. The factor is bounded to $(0.5, 2.0)$ so that the adaptive component remains a controlled adjustment, preventing both near-zero perturbation and excessive amplification.

The structural state adapts the diffusion trajectory, while V provides the history-conditioned prediction-space variance field. Let B_t and $\widetilde{B}_t$ denote cumulative coefficients induced by the adaptive schedule. Appendix B.3 derives these coefficients from the step-wise heteroscedastic variance recursion, giving

$$\nu_t = (B_t - \widetilde{B}_t) \odot V + \widetilde{B}_t \odot S_{\mathrm{tar}} \tag{11}$$

where S_{tar} is a target-dependent empirical variance reference constructed from the observed training target.

The two terms in Eq. (11) play different roles. The history-conditioned variance V is computed solely from the observed historical window and is available at inference, whereas S_{tar} is used only during training as a target-dependent variance reference.

Forecasting additionally provides a predictable component of the future trajectory. Let $\widehat{Y}_0$ denote a deterministic forecast from the historical window. HyperDiff centers the forward process on this

trajectory:

$$Y_t = \sqrt{\bar{\alpha}_t^{\mathrm{adp}}} \odot Y + \left(1 - \sqrt{\bar{\alpha}_t^{\mathrm{adp}}}\right) \odot \widehat{Y}_0 + \sqrt{\nu_t} \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I) \tag{12}$$

This centering separates the history-predictable component from the stochastic deviation modeled by diffusion. In this formulation, $\widehat{Y}_0$ provides the deterministic center, V characterizes history-conditioned predictive variance, and the adaptive schedule controls how stochasticity accumulates along the diffusion trajectory.

At inference, the future Y and the target-dependent reference S_{tar} in the training-time forward marginal are unavailable. We therefore use $\widehat{Y}_0$ as a plug-in estimate of Y and V as the plug-in reference for S_{tar}. Applying these substitutions to Eq. (12) at the terminal diffusion step gives

$$Y_{T_d} = \widehat{Y}_0 + \sqrt{B_{T_d} \odot V} \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I) \tag{13}$$

Here, $B_{T_d} = 1 - \bar{\alpha}_{T_d}$ is determined by the adaptive schedule and is available from the historical condition. This construction initializes reverse diffusion without target-dependent information; the detailed derivation is provided in Appendix B.3.

The influence of $H^\star$ extends beyond forward scheduling. During reverse generation, the refined structure and current adaptive schedule are included in the denoising condition, allowing high-order structure to affect both variable-specific perturbation and recovery from the noisy state. The accumulated variance and reverse-variance target are derived in Appendix B.3, while the exact scheduler, denoising condition, and reverse transition are given in Appendix C.1 and Appendix C.2.

3.4 Training and Probabilistic Inference

With the structural and diffusion conditions defined above, HyperDiff is trained end-to-end with

$$\mathcal{L} = \mathcal{L}_{\mathrm{diff}} + \mathcal{L}_{\mathrm{cond}} + \lambda_{\mathrm{vol}}\mathcal{L}_{\mathrm{vol}} + \lambda_{\mathrm{str}}\mathcal{L}_{\mathrm{str}} \tag{14}$$

where $\mathcal{L}_{\mathrm{diff}} = \mathcal{L}_\epsilon + \lambda_\sigma \mathcal{L}_\sigma$ The objective jointly learns the deterministic center, history-conditioned variance, diffusion denoising, and structure–schedule regularization. Exact loss definitions and weights are provided in Appendix C.2.

At inference, all structural, variance, and scheduling conditions are constructed from the observed history, without using S_{tar}. The reverse process starts from the history-only initialization in Eq. (13) and generates future trajectories through iterative denoising. Repeated sampling with independent Gaussian draws yields an empirical approximation to $p_\theta(Y \mid X_{1:L})$. Complete training and sampling procedures are given in Appendix C.3.

4 Experiments

4.1 Basic Setup

Data and metrics. We evaluate HyperDiff on ETTh1, ETTh2, ETTm1, ETTm2, Exchange Rate, Traffic, and Electricity over horizons $\{96, 192, 336, 720\}$. CRPS and QICE evaluate probabilistic accuracy and calibration, respectively, while Energy Score and Variogram Score assess multivariate joint-distribution quality. Dataset statistics and metric definitions are given in Appendix D.1.

Baselines and implementation. We compare with NsDiff (Ye et al., 2025), KOONPRO (Zheng et al., 2025), TMDM (Li et al., 2024), DiffusionTS (Yuan & Qiao, 2024), FlowTS (Hu et al., 2024), and TimeDiff (Shen & Kwok, 2023). Results are averaged over three random seeds; complete experimental details are provided in Appendix D.1.

4.2 Main Results

Overall probabilistic forecasting performance. HyperDiff achieves the lowest average CRPS and QICE across all seven datasets, indicating better distributional accuracy and calibration. These gains are generally consistent across the four prediction horizons, with relatively small standard errors

Table 1: Average CRPS and QICE (%) over horizons $\{96, 192, 336, 720\}$; lower is better. Full horizon-wise results are reported in Appendix F, Table F.1.

Metric	Model	ETTh1	ETTh2	ETTm1	ETTm2	Exch.	Traffic	Elec.
CRPS↓	HyperDiff (Ours)	**0.387**	**0.349**	**0.341**	**0.293**	**0.342**	**0.370**	**0.294**
	NsDiff (2025)	0.418	0.362	0.361	0.340	0.410	0.378	0.304
	KOONPRO (2025)	0.767	1.672	0.766	1.596	1.449	0.774	0.800
	TMDM (2024)	0.465	0.393	0.361	0.334	0.470	0.562	0.495
	DiffusionTS (2024)	0.615	1.162	0.564	1.072	1.337	0.718	0.651
	FlowTS (2024)	0.629	1.253	0.633	1.150	1.266	0.635	0.482
	TimeDiff (2023)	0.820	0.923	0.738	1.082	1.203	0.772	0.776
QICE (%)↓	HyperDiff (Ours)	**1.305**	**2.242**	**1.707**	**2.188**	**4.230**	**2.349**	**4.887**
	NsDiff (2025)	2.027	2.646	2.082	3.625	6.331	8.211	7.401
	KOONPRO (2025)	14.393	16.853	14.364	15.782	15.511	14.981	15.258
	TMDM (2024)	3.512	3.332	2.875	2.457	6.162	10.597	10.920
	DiffusionTS (2024)	6.541	10.271	4.720	9.958	10.042	13.499	14.291
	FlowTS (2024)	7.146	10.752	10.732	12.572	11.333	5.096	8.997
	TimeDiff (2023)	14.887	15.283	14.894	15.253	15.088	15.189	15.229

in many settings; the averaged results are summarized in Table 1, while complete horizon-wise results are provided in Appendix F, Table F.1. KOONPRO shows a noticeably larger performance gap than the other baselines, which may be related to its oscillatory latent dynamics induced by purely imaginary Koopman eigenvalues and its reliance on a globally aggregated Neural Process summary S for cross-variable information. Its CRPS also changes little across horizons on Traffic and Electricity, while QICE remains consistently high, as shown in Table F.1, suggesting persistent calibration difficulty rather than isolated failures. Overall, the results support the effectiveness of HyperDiff's high-order structure-driven diffusion design.

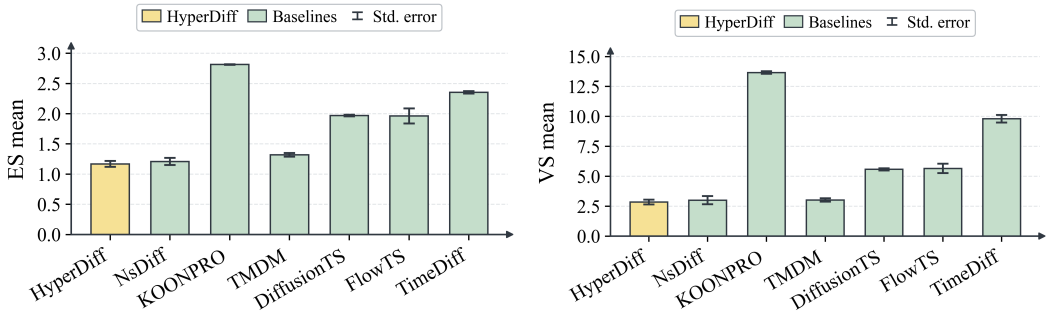
Joint distribution quality. We next evaluate whether HyperDiff improves the multivariate future joint distribution, rather than only marginal accuracy. Figure 3(a) and Figure 3(b) report Energy Score (ES) and Variogram Score (VS), respectively. HyperDiff achieves the lowest mean value on both metrics with small error ranges. Its samples are therefore closer to the true future distribution and better preserve cross-variable dependence.

This result is consistent with the model design. HyperDiff does not generate variables independently; it models coordinated variation and shared uncertainty through dynamic high-order structure learning, volatility propagation, and structure-adaptive diffusion scheduling. Baselines without high-order constraints or structure-aware generation may approximate marginal distributions, but remain limited in capturing joint dependence patterns. The advantage of HyperDiff therefore extends beyond univariate accuracy to multivariate joint-distribution quality.

Computational efficiency. HyperDiff introduces moderate additional computation for dynamic high-order structure learning and structure-adaptive scheduling, while achieving lower CRPS on the representative datasets. Compared with larger generative baselines, it remains substantially more parameter-efficient. Detailed comparisons of accuracy, computational cost, and model complexity are provided in Appendix G.

4.3 Model Analysis

Ablation study. We evaluate the main components of HyperDiff on ETTh1, ETTh2, ETTm1, and Exchange Rate. Removing or replacing the hypergraph, suppressing volatility propagation, replacing the one-step structure–volatility feedback with late fusion, or removing the adaptive scheduler generally degrades CRPS and QICE, as shown in Table 2. These results support the contribution of the proposed structural and stochastic components. Exchange Rate is the main exception, where removing the scheduler slightly improves performance, possibly because its low dimensionality (N=8) provides less structural variation for differentiated noise scheduling.



(a) Energy Score **(b)** Variogram Score

Figure 3: Joint-distribution quality on ETTh1 with prediction horizon 96. (a) Energy Score and (b) Variogram Score; lower is better, and error bars show standard errors.

Table 2: Ablation results on four datasets at prediction horizon 96, reported as mean ± standard error over three runs; lower is better.

Metric	Variant	ETTh1	ETTh2	ETTm1	Exchange Rate
CRPS↓	w/o Hypergraph Structure Learning	0.367±0.003	0.353±0.006	0.332±0.005	0.213±0.006
	w/o Volatility Propagation	0.371±0.008	0.363±0.007	0.332±0.001	0.290±0.006
	w/o Adaptive Diffusion Scheduler	0.373±0.012	0.360±0.003	0.332±0.004	**0.188±0.004**
	Late Fusion instead of One-step Feedback	0.377±0.008	0.369±0.012	0.325±0.009	0.208±0.008
	Graph instead of Hypergraph	0.378±0.009	0.361±0.010	0.323±0.005	0.297±0.004
	HyperDiff (Ours)	**0.364±0.005**	**0.294±0.011**	**0.296±0.001**	0.200±0.004
QICE (%)↓	w/o Hypergraph Structure Learning	1.691±0.306	2.407±0.291	2.908±0.785	4.273±0.653
	w/o Volatility Propagation	2.051±0.092	2.470±0.056	1.970±0.209	4.377±0.109
	w/o Adaptive Diffusion Scheduler	1.332±0.193	2.317±0.277	2.155±0.003	**3.906±0.002**
	Late Fusion instead of One-step Feedback	1.755±0.198	2.505±0.254	2.720±0.319	4.508±0.004
	Graph instead of Hypergraph	1.458±0.178	2.412±0.307	2.646±0.433	4.440±0.402
	HyperDiff (Ours)	**1.216±0.171**	**1.711±0.159**	**1.961±0.365**	4.082±0.385

Structure-aware adaptive diffusion scheduling. We examine whether the learned noise schedule uses structural information. Figure 4 shows that adaptive noise generally increases across diffusion stages and becomes higher in high-volatility states. The scheduler therefore responds to both the diffusion stage and the uncertainty level. In the middle and late stages, structurally central and background nodes also show different distributions: central nodes are more concentrated, whereas background nodes are more dispersed. These observations indicate that HyperDiff combines structural information with volatility states to assign node-specific diffusion trajectories rather than using one fixed schedule.

Hyperparameter sensitivity analysis. We examine whether HyperDiff depends on a narrow hyperparameter choice in dynamic hypergraph construction. We vary the Top-K sparsity level and the Gumbel temperature τ_g. Top-K determines how many hyperedge assignments are retained for each variable, while τ_g controls the smoothness of the Gumbel-softmax assignment. On ETTh1, varying Top-K within $\{1, 2, 5, 8\}$ changes CRPS from 0.361 to 0.371; on Electricity, the range is 0.281 to 0.285. Varying τ_g within $\{0.2, 0.8, 1.0, 2.0\}$ produces similarly small changes. The best setting differs across datasets and metrics, but no single value dominates all cases. These results suggest that HyperDiff is not tuned to a fragile structural configuration. The full sensitivity table is reported in Appendix E.

Residual dependence analysis. HyperDiff selectively reorganizes cross-variable residual dependence rather than uniformly smoothing it. The observed changes align with the learned dynamic high-order structure, with structurally central relations showing stronger dependence weakening. Detailed visualizations and structure-ranked comparisons are provided in Appendix H.

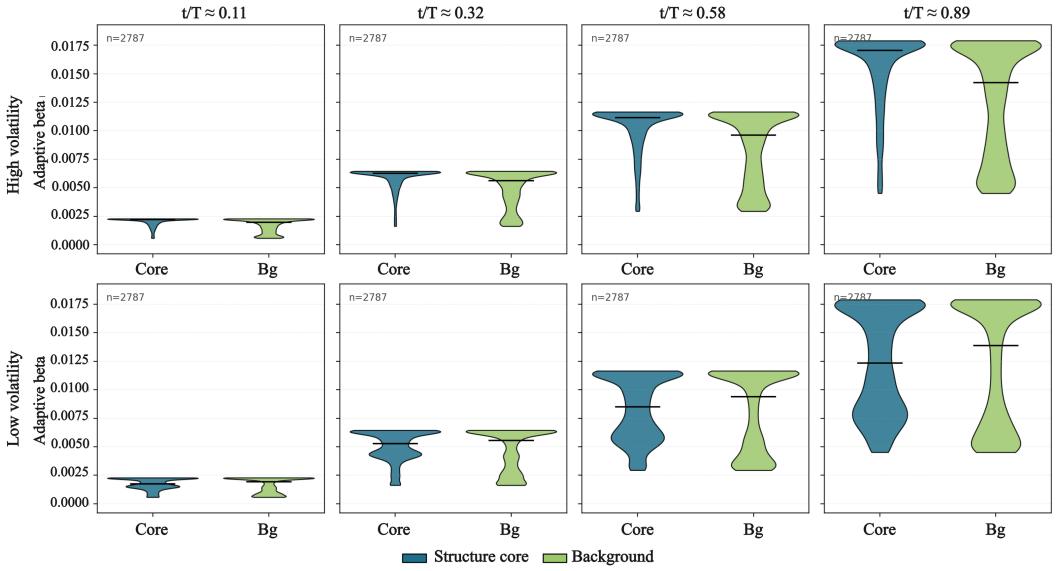


Figure 4: Structure-aware adaptive diffusion scheduling on ETTh1 with prediction horizon 96. Columns show different diffusion stages, rows show high- and low-volatility states, and each panel compares structural-core and background nodes.

Temporal-shift robustness. HyperDiff maintains relatively stable CRPS and QICE under volatility shifts, regime changes, and anomalous windows. Detailed robustness results and baseline comparisons are provided in Appendix I.

This behavior is consistent with the model design. Dynamic hypergraph learning adapts the dependency structure to each input window, while uncertainty propagation and adaptive scheduling condition state-dependent noise injection. Appendix I reports the corresponding baseline curves.

5 Conclusion

We proposed HyperDiff, a high-order structure-driven diffusion framework for multivariate probabilistic forecasting. HyperDiff treats window-conditioned high-order structure as a data-dependent structural prior that organizes history-derived volatility while preserving variable-specific heteroscedasticity. Structural and volatility information then jointly modulate stochasticity during diffusion, allowing dynamic group-wise relations to contribute directly to predictive distribution construction rather than only to representation learning.

Experiments on seven multivariate benchmarks and four forecasting horizons show that HyperDiff improves CRPS and QICE over the evaluated baselines, with consistent evidence from joint-distribution, structural, and robustness analyses. These results suggest that dynamic high-order structure provides an effective bridge between multivariate dependency modeling and predictive uncertainty modeling. Future work will explore more efficient sampling, larger-scale multivariate settings, and probabilistic forecasting with exogenous information.

AI Use Statement

The authors did not use large language models (LLMs) or other AI tools for writing, editing, generating, or developing this manuscript. All aspects of this work, including the proposed methodology, theoretical formulations, experimental design, analysis, and manuscript preparation, were conducted by the authors.

ETHICS STATEMENT

This work studies probabilistic forecasting using publicly available multivariate time-series benchmark datasets. The proposed method does not involve human subjects, personal data collection, or sensitive user information.

The model is developed as a forecasting approach for uncertainty-aware time-series prediction. Although probabilistic forecasting techniques may be applied in domains such as energy, transportation, and finance, the predictions generated by the model should be considered as decision-support information rather than a replacement for domain expertise or human judgment.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. TACTiS-2: Better, faster, simpler attentional copulas for multivariate time series. In *International Conference on Learning Representations*, pp. 22897–22924, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/63796148c99205adb0fcac069cc714d4-Paper-Conference.pdf.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ce1aad92b939420fc17005e5461e6f48-Paper.pdf.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are AllSet: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hpBTIv2uy_E.

Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. TACTiS: Transformer-attentional copulas for time series. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5447–5493. PMLR, 2022. URL https://proceedings.mlr.press/v162/drouin22a.html.

Xinyao Fan, Yueying Wu, Chang Xu, Yuhao Huang, Weiqing Liu, and Jiang Bian. MG-TSD: Multi-granularity time series diffusion models with guided learning process. In *International Conference on Learning Representations*, pp. 12916–12934, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/3792ac815260f4ba55ed71c7906cce36-Paper-Conference.pdf.

Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11979–11987, 2024. doi: 10.1609/aaai.v38i11.29085. URL https://ojs.aaai.org/index.php/AAAI/article/view/29085.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558. URL https://ojs.aaai.org/index.php/AAAI/article/view/4235.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: Deep probabilistic time series imputation. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1651–1661. PMLR, 2020. URL https://proceedings.mlr.press/v108/fortuin20a.html.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014. doi: 10.1146/annurev-statistics-062713-085831.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Yang Hu, Xiao Wang, Zezhen Ding, Lirong Wu, Huatian Zhang, Stan Z. Li, Sheng Wang, Jiheng Zhang, Ziyun Li, and Tianlong Chen. FlowTS: Time series generation via rectified flow, 2024. URL https://arxiv.org/abs/2411.07506.

Van Kwan Zhi Koh, Songnan Lin, Ye Li, Zhiping Lin, and Bihan Wen. IL-DiffTSF: Invertible latent diffusion for probabilistic time series forecasting. *IEEE Internet of Things Journal*, 13(4): 6124–6140, 2026. doi: 10.1109/JIOT.2025.3636872.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28341–28364. Curran Associates, Inc., 2023. doi: 10.52202/075280-1232. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5a1a10c2c2c9b9af1514687bc24b8f3d-Paper-Conference.pdf.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *International Conference on Learning Representations*, pp. 95435–95457, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ee1a1ecc92f35702b5c29dad3dc909ea-Paper-Conference.pdf.

Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *International Conference on Learning Representations*, pp. 67475–67503, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a877dcfbf01491ef9e6dbc31cea4bba2-Paper-Conference.pdf.

Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *International Conference on Learning Representations*, pp. 18604–18622, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/516a9317af9d89e9f2251bd7fde49b8f-Paper-Conference.pdf.

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*, pp. 11116–11140, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/2ea18fdc667e0ef2ad82b2b4d65147ad-Paper-Conference.pdf.

Zhenkai Qin, Baozhong Wei, Qining Luo, Dongze Wu, Weiqi Qin, Xiaochuan Yu, and Ziqian Lin. Modeling carbon price dynamics driven by external factors: A hybrid framework combining mamba state-space model and transformer. *Engineering Applications of Artificial Intelligence*, 167:113725, 2026. doi: 10.1016/j.engappai.2026.113725.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 2021a. URL https://proceedings.mlr.press/v139/rasul21a.html.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=WiGQBFuVRv.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, Sahil Garg, Alexandre Drouin, Nicolas Chapados, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards foundation models for time series forecasting. In *NeurIPS 2023 Workshop on Robustness of Few-shot and Zero-shot Learning in Large Foundation Models (R0-FoMo)*, 2023. URL https://openreview.net/forum?id=jYluzCLFDM.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Zongjiang Shang, Ling Chen, Binqing Wu, and Dongliang Cui. Ada-MSHyper: Adaptive multi-scale hypergraph transformer for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 33310–33337. Curran Associates, Inc., 2024. doi: 10.52202/079017-1050. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3a6935d11910d6f9142b0a1e36fc6753-Paper-Conference.pdf.

Lifeng Shen and James T. Kwok. Non-autoregressive conditional diffusion models for time series prediction. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31016–31029. PMLR, 2023. URL https://proceedings.mlr.press/v202/shen23d.html.

Lifeng Shen, Weiyu Chen, and James T. Kwok. Multi-resolution diffusion models for time series forecasting. In *International Conference on Learning Representations*, pp. 49042–49060, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d64740dd69bcc90ba225a182984b81ba-Paper-Conference.pdf.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, pp. 34635–34667, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/558d48c1f08675daa636e09bfe94a89e-Paper-Conference.pdf.

Edgardo Solano Carrillo, Sreerag Vadakkemeppully Naveenachandran, and Julia Niebling. TEDM: Time series forecasting with elucidated diffusion models. In *International Conference on Learning Representations*, pp. 6696–6726, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0ba40a7aaea28fb9b4fc4a2229fddb71-Paper-Conference.pdf.

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*, pp. 38626–38652, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/a7ac8a21e5a27e7ab31a5f42a0117bdb-Paper-Conference.pdf.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

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. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, 2019. doi: 10.24963/ijcai.2019/264. URL https://www.ijcai.org/proceedings/2019/264.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 753–763. Association for Computing Machinery, 2020. doi: 10.1145/3394486.3403118.

Tijin Yan, Hengheng Gong, Yongping He, Yufeng Zhan, and Yuanqing Xia. Probabilistic time series modeling with decomposable denoising diffusion model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 55759–55777. PMLR, 2024. URL https://proceedings.mlr.press/v235/yan24b.html.

Yanfeng Yang, Siwei Chen, Pingping Hu, Zhaotong Shen, Yingjie Zhang, Zhuoran Sun, Shuai Li, Ziqi Chen, and Kenji Fukumizu. Conditionally whitened generative models for probabilistic time series forecasting. In *International Conference on Learning Representations*, pp. 53263–53298, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/57587d8d6a7ede0e5302fc22d0878c53-Paper-Conference.pdf.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72112–72130. PMLR, 2025. URL https://proceedings.mlr.press/v267/ye25i.html.

Nan Yin, Li Shen, Huan Xiong, Bin Gu, Chong Chen, Xian-Sheng Hua, Siwei Liu, and Xiao Luo. Messages are never propagated alone: Collaborative hypergraph neural network for time-series forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2333–2347, 2024. doi: 10.1109/TPAMI.2023.3331389.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, pp. 41582–41610, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b5b66077d016c037576cc56a82f97f66-Paper-Conference.pdf.

Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryeHuJBtPH.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Tianlong Zhao, Guangle Song, Xuemei Li, Lizhen Cui, and Caiming Zhang. Diff-MGR: Dynamic causal graph attention and pattern reproduction guided diffusion model for multivariate time series probabilistic forecasting. *Information Sciences*, 675:120742, 2024. doi: 10.1016/j.ins.2024.120742.

Ronghua Zheng, Hanru Bai, and Weiyang Ding. KooNPro: A variance-aware koopman probabilistic model enhanced by neural process for time series forecasting. In *International Conference on Learning Representations*, pp. 29457–29481, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4846257e355f6923fc2a1fbe35099e91-Paper-Conference.pdf.

Vincent Zhihao Zheng and Lijun Sun. Multivariate probabilistic time series forecasting with correlated errors. In *Advances in Neural Information Processing Systems*, volume 37, pp. 54288–54329. Curran Associates, Inc., 2024. doi: 10.52202/079017-1720. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/619b8e3ead58dce90bc615f2a7d5d102-Paper-Conference.pdf.

Shiyang Zhou, Zehao Gu, Yun Xiong, Yang Luo, Qiang Wang, and Xiaofeng Gao. REDI: Recurrent diffusion model for probabilistic time series forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3505–3514. Association for Computing Machinery, 2024. doi: 10.1145/3627673.3679808.

A Related Work

Parametric and Distributional Forecasting Models. Early probabilistic forecasting methods typically assume a predefined distribution for future observations and use neural networks to estimate its parameters. DeepAR Salinas et al. (2020), for example, models predictive uncertainty through parameterized likelihoods, but its expressiveness is limited by the chosen distribution family. Later work introduces latent-variable, normalizing-flow, and copula-based models for more flexible predictive distributions. GP-VAE models temporal uncertainty with stochastic latent variables Fortuin et al. (2020), whereas flow-based methods learn flexible distributional transformations Rasul et al. (2021b). TACTiS Drouin et al. (2022) improves modeling of complex multivariate joint distributions, and TACTiS-2 simplifies and accelerates attentional copulas Ashok et al. (2024).

Recent pretrained forecasters explore cross-dataset transfer with Lag-Llama Rasul et al. (2023). Chronos uses tokenized time-series pretraining Ansari et al. (2024), while Moirai targets universal forecasting across frequencies and variate counts Woo et al. (2024). Time-MoE studies billion-scale time-series foundation models with sparse mixture-of-experts routing Shi et al. (2025). These developments increase distributional flexibility and provide a foundation for modern generative probabilistic forecasting.

Diffusion-based Probabilistic Forecasting. Diffusion models learn conditional future distributions through progressive perturbation and reverse denoising Ho et al. (2020). They have become an important approach to probabilistic time-series forecasting. TimeGrad introduces conditional diffusion for multivariate forecasting Rasul et al. (2021a). Diffusion-TS develops an interpretable diffusion model for time-series generation Yuan & Qiao (2024), and TimeDiff proposes a nonautoregressive diffusion forecaster Shen & Kwok (2023). TMDM strengthens conditional diffusion for multivariate probabilistic forecasting Li et al. (2024), while TSDiff explores self-guided diffusion refinement Kollovieh et al. (2023). MG-TSD guides denoising with multiple temporal granularities Fan et al. (2024), mr-Diff uses coarse-to-fine denoising across resolutions Shen et al. (2024), and LDT performs forecasting in a latent diffusion space Feng et al. (2024). REDI combines recurrent modeling with diffusion Zhou et al. (2024).

D^3U separates deterministic and uncertain components before conditional diffusion Li et al. (2025), while TSFlow replaces a fixed generative prior with Gaussian-process-informed flow matching Kollovieh et al. (2025). TEDM adapts elucidated diffusion to temporal forecasting and targets lower sampling complexity Solano Carrillo et al. (2026). IL-DiffTSF applies conditional diffusion in an invertible latent space and directly models forecast uncertainty Koh et al. (2026). Recent work also adapts diffusion to time-varying uncertainty and multivariate dependence. NsDiff introduces nonstationary noise for state-dependent predictive characteristics Ye et al. (2025), Diff-MGR uses dynamic relational information for multivariate generation Zhao et al. (2024), and CW-Gen incorporates conditional mean and covariance information Yang et al. (2026). These methods improve the handling of nonstationarity, heteroscedasticity, and cross-variable dependence. Their conditioning signals, however, mainly come from predictive statistics, covariance, or pairwise relational information. Dynamic high-order group organization as a structural prior for predictive uncertainty remains less explored.

Multivariate Dependency Modeling. Learning variable dependencies is central to multivariate time-series forecasting. Attention-based methods learn data-dependent interactions, whereas graph neural networks organize cross-variable information exchange with explicit graph structures. Graph WaveNet, MTGNN, and AGCRN use adaptive or learnable graphs to improve multivariate representations. Hybrid state-space and attention architectures have also been applied to factor-driven financial forecasting, such as carbon-price modeling with external variables Qin et al. (2026). Graph WaveNet uses adaptive graph convolutions Wu et al. (2019), MTGNN learns graph structure jointly with temporal dynamics Wu et al. (2020), and AGCRN adapts graph convolution to node-specific patterns Bai et al. (2020). Crossformer models cross-dimension dependence with two-stage attention Zhang & Yan (2023), iTransformer uses variate tokens for multivariate correlations Liu et al. (2024), TimeMixer models complementary fine- and coarse-scale patterns Wang et al. (2024), and TimesNet represents periodic variation through a two-dimensional transformation Wu et al. (2023).

Later studies introduce dynamic and input-dependent relation learning, allowing dependencies to vary with time and system state. These methods show the value of explicit structure for multivariate forecasting. However, conventional graphs represent relations primarily through pairwise

edges. Group-wise interactions involving multiple variables are therefore represented only indirectly through collections of pairwise connections.

High-order Structural Modeling. High-order structures represent group-wise dependencies more directly by allowing multiple nodes to participate in one relation. General hypergraph methods such as HGNN Feng et al. (2019), Hyper-SAGNN Zhang et al. (2020), and AllSet Chien et al. (2022) learn high-order representations through hypergraph convolution, attention, or set-based operations. Recent time-series methods use dynamic hypergraphs to characterize group-wise interactions that vary with the historical state Yin et al. (2024). Adaptive multiscale hypergraph learning provides another way to model changing group-wise relations Shang et al. (2024).

These studies show that high-order structures can complement pairwise dependencies and provide richer context for cross-variable interaction. However, existing high-order time-series models mainly use learned structures for representation enhancement, message passing, or feature construction. HyperDiff treats high-order structure as a data-dependent prior for organizing history-derived volatility. The resulting structural information also adapts stochasticity during diffusion. HyperDiff therefore extends high-order dependency modeling from predictive representation to predictive uncertainty organization and distribution construction.

B Derivation of HyperDiff

The main paper introduces the core formulations of HyperDiff. This appendix establishes two mathematical connections behind those designs. First, we show why the preliminary high-order structure naturally corrects a local representation instead of determining volatility on its own. Second, we derive how variable-specific perturbations accumulate into the heteroscedastic forward variance in Section 3.3 and relate this recursion to the reverse-variance target. Exact neural parameterizations and the implemented reverse sampler are given in Appendix C.

B.1 Structure-Regularized Volatility

Let $H = H^{(0)} \in \mathbb{R}_+^{N\times E}$ denote the preliminary incidence matrix introduced in Section 3.1. Define the node and hyperedge degrees by

$$d_i = \sum_e h_{ie} \qquad d_e = \sum_i h_{ie} \tag{B.1}$$

and let D_v and D_e be the corresponding diagonal degree matrices. For a node-level representation $U = [u_1, \dots, u_N]^\top$, the normalized structural energy induced by H is

$$\mathcal{R}_H(U) = \frac{1}{2}\sum_e \frac{1}{d_e}\sum_{i,j} h_{ie}h_{je}\left\|\frac{u_i}{\sqrt{d_i}} - \frac{u_j}{\sqrt{d_j}}\right\|_2^2 \tag{B.2}$$

Expanding the squared differences gives

$$\mathcal{R}_H(U) = \mathrm{tr}(U^\top L_H U) \qquad L_H = I - \Theta_H \qquad \Theta_H = D_v^{-1/2} H D_e^{-1} H^\top D_v^{-1/2} \tag{B.3}$$

The formulation in Section 3.2 can therefore be written as

$$\mathcal{J}(U) = \frac{1}{2}\|U - G_{\mathrm{loc}}\|_F^2 + \frac{\lambda}{2}\mathcal{R}_H(U) \tag{B.4}$$

The first term preserves evidence from each variable's history, whereas the second penalizes differences that are inconsistent with the current high-order organization. Importantly, Eq. (B.4) does not impose identical volatility on variables in the same hyperedge.

Proposition B.1 (Local–structural decomposition). For any $\lambda \geq 0$, Eq. (B.4) has the unique minimizer

$$U^\star = \left[(1+\lambda)I - \lambda\Theta_H\right]^{-1} G_{\mathrm{loc}} \tag{B.5}$$

which satisfies

$$U^\star = (1-\eta)G_{\mathrm{loc}} + \eta\Theta_H U^\star \qquad \eta = \frac{\lambda}{1+\lambda} \tag{B.6}$$

Proof. Since $L_H \succeq 0$, we have $I + \lambda L_H \succ 0$, and Eq. (B.4) is strictly convex. Its stationary condition is

$$\nabla_U \mathcal{J} = U - G_{\text{loc}} + \lambda L_H U = 0 \tag{B.7}$$

which gives

$$(I + \lambda L_H)U^\star = G_{\text{loc}} \tag{B.8}$$

Substituting $L_H = I - \Theta_H$ yields

$$[(1 + \lambda)I - \lambda\Theta_H]U^\star = G_{\text{loc}} \tag{B.9}$$

and hence Eq. (B.5). Rearranging the same equality gives Eq. (B.6). □

The fixed-point form is more informative for the implemented model than the closed-form inverse. It shows that structural information corrects the history-derived local representation rather than replacing it with a structure-determined estimate.

Corollary B.1 (One-step structural correction). Starting Eq. (B.6) from $U^{(0)} = G_{\text{loc}}$ gives

$$U^{(1)} = (1 - \eta)G_{\text{loc}} + \eta\Theta_H G_{\text{loc}} \tag{B.10}$$

Equation (B.10) makes two properties explicit. First, the original local representation remains present after structural regularization. Second, further fixed-point iterations increase the influence of group consistency. This may be undesirable because variables with shared high-order context can still have different local heteroscedasticity. HyperDiff therefore uses the one-step local–structural decomposition instead of repeatedly smoothing the volatility representation toward the exact fixed point.

B.2 Learnable Realization and One-Step Structural Refinement

Corollary B.1 assumes a fixed linear operator Θ_H and a single interpolation coefficient η shared by all variables and forecast positions. HyperDiff relaxes both assumptions. Let $\mathcal{P}_{H^{(0)}}$ denote a learnable node–hyperedge–node transformation constructed from the preliminary structure. The implemented local–structural representation is

$$G_{\text{int}} = A \odot G_{\text{loc}} + (1 - A) \odot \mathcal{P}_{H^{(0)}}(G_{\text{loc}}) \tag{B.11}$$

Here, $\mathcal{P}_{H^{(0)}}$ generalizes the fixed operator Θ_H, whereas $1 - A$ generalizes the scalar coefficient η into a sample-, variable-, and horizon-dependent correction strength. The model retains the local–structural interpretation of Corollary B.1 without assuming linear propagation or uniform smoothing.

The integrated representation produces

$$V_{\text{int}} = \text{Softplus}(G_{\text{int}}^\top) \qquad F^{\text{vol}} = \Phi_{\text{vol}}(G_{\text{int}}) \tag{B.12}$$

V_{int} remains aligned with the forecasting horizon, while F^{vol} maps the same state into latent structural space. Because the preliminary structure is inferred before this structure-conditioned volatility is available, we use the feedback for one structural recalibration:

$$H^\star = f_H(X_{1:L}; F^{\text{vol}}) \tag{B.13}$$

This feedback is deliberately noniterative. $H^{(0)}$ organizes the history-derived volatility once, and the resulting state then refines the high-order structure once. The refined $H^\star$ is not fed back to recompute V_{int}, which preserves the structure–volatility coupling without repeatedly increasing structural smoothing.

HyperDiff also retains a direct temporal variance estimate V_{ext} outside the hypergraph branch. The final history-conditioned variance field is

$$V = \gamma \odot V_{\text{ext}} + (1 - \gamma) \odot V_{\text{int}} \tag{B.14}$$

The two gates operate at different levels. A determines how strongly structural context corrects the local volatility representation, whereas γ balances the structure-conditioned and direct temporal estimates in the final scale. Exact parameterizations are given in Appendix C.1.

B.3 CENTERED HETEROSCEDASTIC DIFFUSION

The final variance field V characterizes the history-conditioned predictive variance over future positions and variables. In contrast, the adaptive schedule in Section 3.3 controls how stochasticity is introduced along the diffusion trajectory. Let

$$\alpha_t = 1 - \beta_t^{\mathrm{adp}} \qquad \bar{\alpha}_t = \prod_{r=1}^{t} \alpha_r \tag{B.15}$$

For clarity, we suppress batch, forecast-position, and variable indices in this subsection. All variance operations are elementwise. We index the diffusion steps by $t = 1, \ldots, T_d$.

HyperDiff perturbs the future around the deterministic history-conditioned forecast $\widehat{Y}_0$. Equivalently, the forward state is

$$Y_t = \widehat{Y}_0 + \sqrt{\bar{\alpha}_t}(Y - \widehat{Y}_0) + \sqrt{\nu_t}\,\epsilon \qquad \epsilon \sim \mathcal{N}(0, I) \tag{B.16}$$

Thus, diffusion acts on deviations from the predictable center. The remaining question is how the accumulated heteroscedastic variance ν_t depends on V, the target-dependent training reference, and the adaptive schedule.

During training, let S_{tar} denote the target-dependent empirical variance defined in Appendix C.2. The variance contribution at step t is

$$\Lambda_t = (1 - \alpha_t)^2 V + \alpha_t(1 - \alpha_t) S_{\mathrm{tar}} \tag{B.17}$$

The corresponding HyperDiff recursion is

$$\nu_t = \alpha_t \nu_{t-1} + \Lambda_t \qquad \nu_0 = 0 \tag{B.18}$$

Proposition B.2 (Closed-form heteroscedastic variance). Define

$$B_t = 1 - \bar{\alpha}_t \tag{B.19}$$

and

$$\widetilde{B}_t = \sum_{s=1}^{t} (1 - \alpha_s) \prod_{r=s}^{t} \alpha_r \tag{B.20}$$

Then the solution of Eq. (B.18) is

$$\nu_t = (B_t - \widetilde{B}_t) V + \widetilde{B}_t S_{\mathrm{tar}} \tag{B.21}$$

Proof. Equation (B.20) satisfies

$$\widetilde{B}_t = \alpha_t \widetilde{B}_{t-1} + \alpha_t(1 - \alpha_t) \tag{B.22}$$

while $B_t = 1 - \alpha_t \bar{\alpha}_{t-1}$ gives

$$B_t = \alpha_t B_{t-1} + (1 - \alpha_t) \tag{B.23}$$

Subtracting the two recursions yields

$$B_t - \widetilde{B}_t = \alpha_t (B_{t-1} - \widetilde{B}_{t-1}) + (1 - \alpha_t)^2 \tag{B.24}$$

Assume that

$$\nu_{t-1} = (B_{t-1} - \widetilde{B}_{t-1}) V + \widetilde{B}_{t-1} S_{\mathrm{tar}} \tag{B.25}$$

Substituting this expression and Eq. (B.17) into Eq. (B.18) gives

$$\nu_t = \left[\alpha_t (B_{t-1} - \widetilde{B}_{t-1}) + (1 - \alpha_t)^2\right] V + \left[\alpha_t \widetilde{B}_{t-1} + \alpha_t(1 - \alpha_t)\right] S_{\mathrm{tar}} \tag{B.26}$$

Applying Eqs. (B.22)–(B.24) gives Eq. (B.21). The result follows by induction from $B_0 = \widetilde{B}_0 = \nu_0 = 0$. □

Proposition B.2 shows that the forward variance in the main paper is not an independent interpolation rule. It is the closed form of the recursion induced by the adaptive schedule. The coefficient $B_t - \widetilde{B}_t$ accumulates the contribution from the history-conditioned variance field V, whereas $\widetilde{B}_t$ accumulates the contribution from the target-observed training reference S_{tar}.

History-only terminal initialization. At inference, the target-dependent quantities used by the training-time forward process are unavailable. We therefore substitute the deterministic forecast $\widehat{Y}_0$ for the unknown future Y and the history-conditioned variance V for the target-dependent reference S_{tar} in the terminal form of the forward marginal. Because $B_{T_d} = 1 - \bar{\alpha}_{T_d}$, these plug-in substitutions give

$$Y_{T_d}^{\mathrm{hist}} = \widehat{Y}_0 + \sqrt{B_{T_d} \odot V} \odot \epsilon \qquad \epsilon \sim \mathcal{N}(0, I) \tag{B.27}$$

This is the history-only initialization stated in Eq. (13). Its remaining approximation error is associated with the deterministic forecast residual and the difference between V and S_{tar}; both are addressed by the corresponding training objectives.

Substituting Eq. (B.21) into Eq. (B.16) yields

$$Y_t = \sqrt{\bar{\alpha}_t} Y + (1 - \sqrt{\bar{\alpha}_t})\widehat{Y}_0 + \sqrt{(B_t - \widetilde{B}_t)V + \widetilde{B}_t S_{\mathrm{tar}}}\, \epsilon \tag{B.28}$$

This expression clarifies the roles of the two adaptive quantities. V determines the history-conditioned scale of predictive variation, whereas $\{\alpha_t\}$ determines how that variation accumulates along the diffusion trajectory.

Proposition B.3 (Reverse-variance target). Let

$$\sigma_{1,t} = \Lambda_t = (1 - \alpha_t)^2 V + \alpha_t (1 - \alpha_t) S_{\mathrm{tar}} \tag{B.29}$$

be the variance contribution introduced at the current step, and let

$$\sigma_{2,t} = \nu_{t-1} = (B_{t-1} - \widetilde{B}_{t-1})V + \widetilde{B}_{t-1} S_{\mathrm{tar}} \tag{B.30}$$

be the accumulated variance before that step. For a diagonal linear Gaussian update with coefficient $\sqrt{\alpha_t}$, the conditional reverse variance is

$$\widetilde{\sigma}_t = \frac{\sigma_{1,t}\sigma_{2,t}}{\alpha_t \sigma_{2,t} + \sigma_{1,t}} \tag{B.31}$$

Proof. For an individual forecast position and variable, Gaussian conditioning gives the posterior precision

$$\widetilde{\sigma}_t^{-1} = \sigma_{2,t}^{-1} + \alpha_t \sigma_{1,t}^{-1} \tag{B.32}$$

Taking its inverse yields

$$\widetilde{\sigma}_t = \left(\sigma_{2,t}^{-1} + \alpha_t \sigma_{1,t}^{-1}\right)^{-1} = \frac{\sigma_{1,t}\sigma_{2,t}}{\alpha_t \sigma_{2,t} + \sigma_{1,t}} \tag{B.33}$$

Since HyperDiff uses diagonal variance parameterizations over forecast positions and variables, the identity applies elementwise. □

The reverse-variance target for training $\sigma_{\theta,t}$ is consistent with the variance decomposition in Proposition B.2. The implemented reverse mean also uses a denoiser-based clean-state reconstruction and an auxiliary variance recovered from $\sigma_{\theta,t}$. These sampling-specific quantities are given in Appendix C.2, separately from the target-observed forward derivation.

C Implementation and Optimization Details

Appendix B explains the mathematical basis of the main modeling choices. This section gives the exact neural parameterizations, optimization objective, and reverse-sampling procedure required for reproduction.

C.1 Exact Structural, Volatility, and Diffusion Parameterization

Sparse window-conditioned structure. For variable i, the structural learner combines the node state, projected local descriptors, and optional volatility feedback:

$$c_i = \phi_c([z_i; \widetilde{s}_i; f_i^{\mathrm{vol}}]) \tag{C.1}$$

where $f_i^{\text{vol}} = 0$ during preliminary structure learning. Given the affinity a_{ie} in the main paper, the sparse incidence weights are

$$\pi_{ie} = \left[\text{Softmax}\left(\frac{a_i + g_i}{\tau_g}\right)\right]_e \tag{C.2a}$$

$$m_{ie} = \mathbb{I}[e \in \text{TopK}(a_i, K)] \tag{C.2b}$$

$$h_{ie} = m_{ie}\pi_{ie}\,\text{Sigmoid}(a_{ie}) \tag{C.2c}$$

Top-K selection is applied over candidate hyperedges separately for each variable.

Given $H = [h_{ie}]$, direct variable interaction is restricted to pairs that share at least one active hyperedge. Let

$$M_H(i,j) = \mathbb{I}[(HH^\top)_{ij} > 10^{-8}] \tag{C.3}$$

A structural encoding layer can be summarized as

$$\bar{Z} = \text{MHA}(Z; M_H) \tag{C.4a}$$

$$E_H = \Psi_e(D_e^{-1}H^\top \bar{Z}) \tag{C.4b}$$

$$Z^+ = \text{LN}\left(Z + \Phi_n(D_v^{-1}HE_H)\right) \tag{C.4c}$$

The layer is followed by a residual feed-forward block. With multiple structural layers, the incidence matrix can be recomputed from updated node states between layers. $H^{(0)}$ and $H^\star$ denote the final incidence structures returned by the preliminary and volatility-informed encoder passes, respectively.

Learnable volatility correction. The local volatility encoder maps each variable directly from history length L to forecast horizon P:

$$G_{\text{loc}} = f_{\text{loc}}(X_{1:L}) \in \mathbb{R}^{B\times N\times P} \tag{C.5}$$

The operator $\mathcal{P}_{H^{(0)}}$ in Eq. (B.11) is implemented as

$$E_{\text{vol}} = \Psi_e\left(D_e^{-1}H^{(0)\top}\Phi_{n\to e}(G_{\text{loc}})\right) \tag{C.6a}$$

$$G_{\text{prop}} = \Phi_{e\to n}\left(D_v^{-1}H^{(0)}E_{\text{vol}}\right) \tag{C.6b}$$

$$A = \text{Sigmoid}\left(\Phi_A([G_{\text{loc}}; G_{\text{prop}}])\right) \tag{C.6c}$$

For a multivariate sequence $Z \in \mathbb{R}^{T\times N}$, let $\mathcal{V}_R(Z; w)$ denote the trailing rolling-variance operator applied along the temporal dimension with window length w. For variable i and time position t, it computes

$$[\mathcal{V}_R(Z;w)]_{t,i} = \frac{1}{w}\sum_{\tau=t-w+1}^{t}\left(Z_{\tau,i} - \bar{Z}_{t,i}^{(w)}\right)^2 \qquad \bar{Z}_{t,i}^{(w)} = \frac{1}{w}\sum_{\tau=t-w+1}^{t} Z_{\tau,i} \tag{C.7}$$

At the left boundary, replicate padding is used when fewer than w observations are available. The reported configuration uses $w = 96$ and the population-variance estimator, consistent with the implementation.

The direct temporal branch and final fusion gate are

$$V_{\text{ext}} = \text{Softplus}\left(f_{\text{ext}}(\mathcal{V}_R(X_{1:L}; w))\right) \tag{C.8a}$$

$$\gamma = \text{Clip}_{[10^{-4},\,1-10^{-4}]}\left[\text{Sigmoid}\left(\Phi_\gamma([V_{\text{ext}}; V_{\text{int}}])\right)\right] \tag{C.8b}$$

Here, "external" means external to the hypergraph propagation branch; no external observations are used.

Adaptive scheduler. Let $m_{b,i,e} = \mathbb{I}[h^\star_{b,i,e} > 10^{-6}]$ and $q_{b,e} = \sum_j h^\star_{b,j,e}$. The scheduler receives the following structural statistics:

$$d_{b,i} = \sum_e m_{b,i,e} \tag{C.9a}$$

$$\bar{h}_{b,i} = \frac{\sum_e m_{b,i,e} h^\star_{b,i,e}}{\max(d_{b,i}, 1)} \tag{C.9b}$$

$$\ell_{b,i} = \frac{\sum_e h^\star_{b,i,e} q_{b,e}}{\max(d_{b,i}, 1)} \tag{C.9c}$$

$$c_{b,i} = \frac{d_{b,i}}{\max_j d_{b,j}} \tag{C.9d}$$

With $r_{b,i} = [d_{b,i}, \bar{h}_{b,i}, \ell_{b,i}, c_{b,i}]$, the scheduler context is

$$u_{b,t,i} = [z^\star_{b,i}; \widetilde{s}_{b,i}; f^{\mathrm{vol}}_{b,i}; e_t; r_{b,i}] \tag{C.10}$$

The resulting schedule is

$$\beta^{\mathrm{adp}}_{b,t,i} = \mathrm{Clip}_{[10^{-6}, 0.999]}\left(\beta_t \left[0.5 + 1.5\,\mathrm{Sigmoid}(\mathcal{M}(u_{b,t,i}) + \zeta_i)\right]\right) \tag{C.11}$$

Here, ζ_i denotes the learnable variable-specific bias used by the adaptive scheduler.

Structure-conditioned denoising. Let E_X denote the refined encoded history. The implementation forms a node profile and a current schedule profile,

$$p^H_{b,i} = \sum_e h^\star_{b,i,e} \qquad p^\beta_{b,t,i} = \beta^{\mathrm{adp}}_{b,t,i} \tag{C.12}$$

After both profiles are broadcast over the encoded-history length, the denoising condition is

$$\widetilde{C}_t = \Phi_C([E_X; p^H; p^\beta_t]) \tag{C.13a}$$

$$G_t = \mathrm{Sigmoid}(\Phi_G([E_X; \widetilde{C}_t])) \odot \mathrm{Sigmoid}(\phi_t(t/T_d)) \tag{C.13b}$$

$$C_t = (1 - G_t) \odot E_X + G_t \odot \widetilde{C}_t \tag{C.13c}$$

Thus, the refined high-order state influences both the adaptive schedule and the historical condition provided to the reverse denoiser.

C.2 Training Objective and Reverse-Sampling Realization

Throughout this section, $\varepsilon_{\mathrm{num}} > 0$ denotes a small constant used only for numerical stability, whereas $\epsilon \sim \mathcal{N}(0, I)$ denotes Gaussian diffusion noise.

During training, let

$$\bar{X} = [X_{1:L}; Y] \qquad S_{\mathrm{tar}} = \mathcal{V}_R(\bar{X}; w)_{-P:} + \varepsilon_{\mathrm{num}} \tag{C.14}$$

denote the target-dependent empirical variance reference. Because the final P variance estimates depend on Y, S_{tar} is used only during training.

Optimization objective. The primary supervision terms are

$$\mathcal{L}_{\mathrm{cond}} = \left\langle (\widehat{Y}_0 - Y)^2 \right\rangle \tag{C.15a}$$

$$\mathcal{L}_{\mathrm{vol}} = \left\langle (\sqrt{V_{\mathrm{ext}}} - \sqrt{S_{\mathrm{tar}}})^2 \right\rangle + \left\langle (\sqrt{V_{\mathrm{int}} + \varepsilon_{\mathrm{num}}} - \sqrt{S_{\mathrm{tar}}})^2 \right\rangle \tag{C.15b}$$

$$\mathcal{L}_\epsilon = \left\langle (\epsilon - \widehat{\epsilon}_\theta)^2 \right\rangle \tag{C.15c}$$

$$\mathcal{L}_\sigma = \left\langle \frac{\widetilde{\sigma}_t}{\sigma_{\theta,t}} - \log\left(\frac{\widetilde{\sigma}_t}{\sigma_{\theta,t}} + \varepsilon_{\mathrm{num}}\right) \right\rangle \tag{C.15d}$$

The square-root variance loss compares standard-deviation amplitudes rather than raw variances, which reduces the dominance of high-variance variables. It directly supervises the component estimates V_{ext} and V_{int} toward the target-referenced scale, while the fused variance V is obtained from their learned combination.

For structure–schedule consistency, define

$$w_{b,e} = \sum_i h^\star_{b,i,e} \tag{C.16a}$$

$$\bar{\beta}_{b,e,t} = \frac{\sum_i h^\star_{b,i,e} \beta^{\mathrm{adp}}_{b,t,i}}{w_{b,e} + \varepsilon_{\mathrm{num}}} \tag{C.16b}$$

$$\beta^H_{b,t,i} = \frac{\sum_e h^\star_{b,i,e} \bar{\beta}_{b,e,t}}{\sum_e h^\star_{b,i,e} + \varepsilon_{\mathrm{num}}} \tag{C.16c}$$

The corresponding regularization terms are

$$\mathcal{L}_{\mathrm{cons}} = \left\langle (\beta^{\mathrm{adp}} - \beta^H)^2 \right\rangle \tag{C.17a}$$

$$p_{b,i,e} = \frac{h^\star_{b,i,e}}{\sum_j h^\star_{b,j,e} + \varepsilon_{\mathrm{num}}} \tag{C.17b}$$

$$\mathcal{L}_{\mathrm{ent}} = \left\langle -\sum_i p_{b,i,e} \log(p_{b,i,e} + \varepsilon_{\mathrm{num}}) \right\rangle_{b,e} \tag{C.17c}$$

$$\mathcal{L}_{\mathrm{smooth}} = \left\langle (\beta^{\mathrm{adp}}_{t+1} - \beta^{\mathrm{adp}}_t)^2 \right\rangle \tag{C.17d}$$

Because the positive entropy term is minimized with a positive coefficient, it favors lower-entropy, more selective memberships over diffuse assignments.

The effective objective is

$$\mathcal{L}_{\mathrm{str}} = \mathcal{L}_{\mathrm{cons}} + 0.01\mathcal{L}_{\mathrm{ent}} + 0.1\mathcal{L}_{\mathrm{smooth}} \tag{C.18a}$$

$$\mathcal{L} = \mathcal{L}_\epsilon + \mathcal{L}_{\mathrm{cond}} + 0.5\mathcal{L}_{\mathrm{vol}} + 0.5\mathcal{L}_\sigma + 0.05\mathcal{L}_{\mathrm{str}} \tag{C.18b}$$

The implementation can also compute a sample-based calibration quantity. The corresponding reverse samples are generated without gradients, so this quantity does not contribute to the effective objective in Eq. (C.18).

Reverse-sampling realization. We distinguish the deterministic center $\widehat{Y}_0$ from the reverse-process clean reconstruction $\widetilde{Y}_0$. The former is computed directly from the observed historical window before diffusion, whereas the latter is reconstructed from the current noisy diffusion state during reverse generation.

At inference, the target-dependent reference S_{tar} is unavailable and is not reconstructed from future observations. Following the plug-in terminal construction derived in Appendix B.3, we replace the unknown future by $\widehat{Y}_0$ and S_{tar} by V to initialize the reverse process:

$$Y_{T_d} = \widehat{Y}_0 + \sqrt{B_{T_d} \odot V} \odot z \qquad z \sim \mathcal{N}(0, I) \tag{C.19}$$

No target-dependent variance statistic is required for this initialization.

At reverse step t, the denoiser predicts $\widehat{\epsilon}_\theta$ and a positive variance $\sigma_{\theta,t}$. The implementation recovers the auxiliary clean-state variance $\widehat{S}_t$ by solving

$$\lambda_{0,t}\widehat{S}_t^2 + \lambda_{1,t}\widehat{S}_t + \lambda_{2,t} = 0 \qquad \widehat{S}_t = \frac{-\lambda_{1,t} + \sqrt{\lambda_{1,t}^2 - 4\lambda_{0,t}\lambda_{2,t}}}{2\lambda_{0,t}} \tag{C.20}$$

Its coefficients are

$$\lambda_{0,t} = \alpha_t(1-\alpha_t)\widetilde{B}_{t-1} + \varepsilon_{\mathrm{num}} \tag{C.21a}$$

$$\lambda_{1,t} = \left[(1-\alpha_t)^2\widetilde{B}_{t-1} + \alpha_t(1-\alpha_t)(B_{t-1} - \widetilde{B}_{t-1})\right] \odot V - \sigma_{\theta,t}\left[\alpha_t\widetilde{B}_{t-1} + \alpha_t(1-\alpha_t)\right] \tag{C.21b}$$

$$\lambda_{2,t} = V^2(1-\alpha_t)^2(B_{t-1} - \widetilde{B}_{t-1}) - \sigma_{\theta,t} \odot V\left[\alpha_t B_{t-1} - \alpha_t\widetilde{B}_{t-1} + (1-\alpha_t)^2\right] \tag{C.21c}$$

$\widehat{S}_t$ is inferred from the current model state. It is not an estimate of the unavailable target-derived training statistic S_{tar}.

Algorithm 1 Training HyperDiff

Require: Historical window $X_{1:L}$ and observed future Y

Compute the deterministic center $\widehat{Y}_0$.
Infer the preliminary high-order structure $H^{(0)}$.
Construct G_{loc} and its one-step learnable structural correction.
Obtain V_{int} and F^{vol}, and refine the structure once to obtain $H^{\star}$.
Estimate V_{ext} and fuse the two variance estimates to obtain V.
Construct the variable-specific adaptive schedule.
Construct the training-only target-dependent empirical variance reference S_{tar}.
Sample diffusion step t and $\epsilon \sim \mathcal{N}(0, I)$.
Compute ν_t according to Proposition B.2 and generate the noisy state Y_t.
Predict $\widehat{\epsilon}_\theta$ and $\sigma_{\theta,t}$.
Construct $\widetilde{\sigma}_t$ according to Proposition B.3.
Evaluate Eq. (C.18) and update all trainable parameters.

Using the variance decomposition from Proposition B.2,

$$\widehat{\nu}_t = (B_t - \widetilde{B}_t)V + \widetilde{B}_t\widehat{S}_t \tag{C.22}$$

the clean state is reconstructed as

$$\widetilde{Y}_0 = \frac{Y_t - (1 - \sqrt{\bar{\alpha}_t})\widehat{Y}_0 - \widehat{\epsilon}_\theta\sqrt{\widehat{\nu}_t}}{\sqrt{\bar{\alpha}_t} + \varepsilon_{\mathrm{num}}} \tag{C.23}$$

Replacing the training-time S_{tar} with $\widehat{S}_t$ in Eqs. (B.29)–(B.30), define

$$D_t = \alpha_t\sigma_{2,t} + \sigma_{1,t} + \varepsilon_{\mathrm{num}} \tag{C.24}$$

The implemented interpolation coefficients are

$$\gamma_{0,t} = \frac{\sqrt{\bar{\alpha}_{t-1}}\sigma_{1,t}}{D_t} \tag{C.25a}$$

$$\gamma_{1,t} = \frac{\sqrt{\alpha_t}\sigma_{2,t}}{D_t} \tag{C.25b}$$

$$\gamma_{2,t} = \frac{\sqrt{\alpha_t}(\alpha_t - 1)\sigma_{2,t} + (1 - \sqrt{\bar{\alpha}_{t-1}})\sigma_{1,t}}{D_t} \tag{C.25c}$$

The reverse mean and stochastic transition are

$$\mu_{\theta,t} = \gamma_{0,t}\widetilde{Y}_0 + \gamma_{1,t}Y_t + \gamma_{2,t}\widehat{Y}_0 \tag{C.26a}$$

$$Y_{t-1} = \mu_{\theta,t} + \sqrt{\tau_\sigma\sigma_{\theta,t}} \odot z \qquad z \sim \mathcal{N}(0, I) \tag{C.26b}$$

The reported configuration uses $\tau_\sigma = 1.2$. At $t = 0$, we return the clean reconstruction in Eq. (C.23) without additional stochastic perturbation.

C.3 Training and Forecasting Algorithms

The implementation uses antithetic pairing of diffusion steps within each mini-batch to cover different regions of the diffusion trajectory during training.

The target-dependent empirical reference S_{tar} does not appear in Algorithm 2. All structural states, volatility conditions, and adaptive schedules required at inference are constructed only from the observed historical window.

D Experimental Settings and Evaluation Protocol

D.1 Experimental Settings

For horizons 96, 192, 336, and 720, we use an adaptive input-window strategy that selects the look-back length required by each forecasting task. All experiments run on a single Nvidia RTX 3090 GPU with 24GB of memory, using PyTorch 2.1.0 under Ubuntu 22.04. The batch size is 32 and the initial learning rate is 5×10^{-4}. We train all models for 20 epochs with early-stopping patience of 8. Each experiment uses three random seeds. We report the mean and corresponding standard error.

Algorithm 2 Probabilistic Forecasting with HyperDiff

Require: Historical window $X_{1:L}$ and number of trajectories M

Compute the deterministic center $\widehat{Y}_0$.
Infer $H^{(0)}$, construct the structure-conditioned volatility state, and refine the structure to $H^\star$.
Fuse V_{ext} and V_{int} to obtain V.
Construct the adaptive diffusion conditions using history only.
for $m = 1, \ldots, M$ **do**
  Sample $z \sim \mathcal{N}(0, I)$ and initialize $Y_{T_d} = \widehat{Y}_0 + \sqrt{B_{T_d} \odot V} \odot z$.
  for $t = T_d - 1, \ldots, 1$ **do**
    Predict $\widehat{\epsilon}_\theta$ and $\sigma_{\theta,t}$.
    Recover $\widehat{S}_t$ using Eq. (C.20).
    Reconstruct $\widetilde{Y}_0$ using Eq. (C.23).
    Construct $\mu_{\theta,t}$ and sample Y_{t-1} using Eq. (C.26).
  end for
  After the final reverse step at $t = 1$, return the clean reconstruction $\widetilde{Y}_0$.
end for
return M predictive trajectories.

D.2 DATASETS

This section provides additional information about the seven benchmarks. All datasets are chronological multivariate time series, and the train, validation, and test partitions follow the time axis to avoid future-information leakage. We use horizons $\{96, 192, 336, 720\}$ for every dataset, allowing the same protocol to evaluate short- and long-range uncertainty modeling. Table D.1 summarizes the domain, sampling frequency, dimensionality, time span, total length, and split sizes.

- **ETTh1 / ETTh2:** These two hourly subsets come from the Electricity Transformer Temperature (ETT) benchmark and record measurements from two transformer stations over two years. Each subset contains 7 variables: six load-related features (HUFL, HULL, MUFL, MULL, LUFL, and LULL) and the oil temperature variable (OT). The hourly sampling makes these datasets useful for evaluating daily and weekly temporal dependencies.
- **ETTm1 / ETTm2:** These two subsets correspond to the same ETT data sources as ETTh1 and ETTh2 but are sampled every 15 minutes. They also contain the same 7 variables, while the longer sequence length and finer temporal resolution make them more demanding for local pattern modeling and long-horizon generation.
- **Exchange Rate:** This dataset contains daily exchange rates of 8 countries from 1990 to 2016. Compared with the ETT benchmarks, it is low-dimensional but exhibits stronger nonstationarity, abrupt trend changes, and scale variations, making it a useful test case for uncertainty calibration under financial dynamics.
- **Traffic:** This dataset records hourly road occupancy rates collected by 862 sensors on San Francisco Bay Area freeways. Its high dimensionality and spatially correlated sensor readings make it a challenging benchmark for modeling dense cross-variable dependencies and coordinated temporal variation.
- **Electricity:** This benchmark contains hourly electricity consumption from 321 customers between 2012 and 2014. The series contain heterogeneous user-level demand patterns, including daily and weekly seasonality, making the dataset suitable for evaluating probabilistic forecasts under recurring but nonidentical consumption behavior.

D.3 EVALUATION METRICS

We evaluate both marginal probabilistic forecasting quality and multivariate joint-distribution quality. CRPS and QICE are used as the primary metrics, while Energy Score and Variogram Score are used for joint-distribution analysis. Lower values indicate better performance for all reported metrics.

Table D.1: Statistics of datasets

Dataset	Domain	Frequency	Variables	Time Span	Total Length	N_{train}	N_{val}	N_{test}
ETTh1	Electricity transformer	1 hour	7	2016–2018	17,420	12,194	1,742	3,484
ETTh2	Electricity transformer	1 hour	7	2016–2018	17,420	12,194	1,742	3,484
ETTm1	Electricity transformer	15 min	7	2016–2018	69,680	48,776	6,968	13,936
ETTm2	Electricity transformer	15 min	7	2016–2018	69,680	48,776	6,968	13,936
Exchange Rate	Finance	1 day	8	1990–2016	7,588	5,311	758	1,519
Traffic	Transportation	1 hour	862	2015–2016	17,544	12,280	1,754	3,510
Electricity	Energy	1 hour	321	2012–2014	26,304	18,412	2,630	5,262

Continuous Ranked Probability Score. Given generated samples $\{\hat{y}^{(s)}\}_{s=1}^{S}$ and ground truth y, the Monte Carlo estimate of CRPS is

$$\text{CRPS} = \frac{1}{S}\sum_{s=1}^{S}\left|\hat{y}^{(s)} - y\right| - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\left|\hat{y}^{(s)} - \hat{y}^{(s')}\right| \tag{D.1}$$

CRPS is a strictly proper scoring rule that jointly reflects predictive accuracy and sharpness.

Quantile Interval Coverage Error. QICE evaluates calibration by comparing empirical quantile coverage with nominal coverage:

$$\text{QICE} = \frac{1}{K}\sum_{k=1}^{K}\left|\hat{c}_k - \frac{1}{K}\right| \tag{D.2}$$

where $\hat{c}_k$ denotes the empirical proportion of ground-truth values falling into the k-th predicted quantile interval. QICE is reported in percent for readability.

Energy Score. For joint-distribution evaluation, each prediction step is treated as an N-dimensional vector. Let $y \in \mathbb{R}^N$ denote the multivariate ground truth and $\hat{y}^{(s)}$ the corresponding generated sample. The Energy Score is

$$\text{ES} = \frac{1}{S}\sum_{s=1}^{S}\left\|\hat{y}^{(s)} - y\right\|_2 - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\left\|\hat{y}^{(s)} - \hat{y}^{(s')}\right\|_2 \tag{D.3}$$

Variogram Score. We use $p = 0.5$ following the implementation:

$$\text{VS}_p = \sum_{i<j}\left(|y_i - y_j|^p - \frac{1}{S}\sum_{s=1}^{S}\left|\hat{y}_i^{(s)} - \hat{y}_j^{(s)}\right|^p\right)^2 \tag{D.4}$$

Energy Score evaluates multivariate distributional agreement, whereas Variogram Score is particularly sensitive to cross-variable dependence.

E Hyperparameter Sensitivity Analysis

We examine the sensitivity of HyperDiff to two hyperparameters in dynamic hypergraph construction: Top-K and the Gumbel temperature τ_g. Top-K controls the sparsity of the learned high-order structure, while τ_g controls the smoothness of the Gumbel-softmax assignment. As shown in Table E.1, changing Top-K on ETTh1 and Electricity produces only small CRPS variations. QICE fluctuates more, but remains relatively stable overall.

Varying τ_g produces a similar pattern. Different datasets and metrics prefer slightly different values, but no single setting dominates all cases. HyperDiff therefore does not rely on a fragile single hyperparameter configuration and remains stable across the tested sparsity levels and assignment temperatures.

Table E.1: Hyperparameter sensitivity of HyperDiff on ETTh1 and Electricity. We vary Top-K and the Gumbel temperature τ_g while keeping all other settings fixed. Values are mean $\pm$ standard error over three runs; lower is better. The lowest mean value within each sweep is bolded for readability.

Hyperparameter	Value	ETTh1		Electricity	
		CRPS↓	QICE (%)↓	CRPS↓	QICE (%)↓
Default	–	0.361±0.003	1.456±0.211	0.282±0.002	7.827±0.107
Top-K	1	**0.361±0.009**	1.531±0.387	0.282±0.001	7.882±0.058
	2	0.371±0.008	1.508±0.198	0.285±0.001	7.987±0.082
	5	0.363±0.009	**1.445±0.168**	**0.281±0.002**	7.875±0.067
	8	0.366±0.004	1.589±0.211	0.282±0.002	**7.761±0.082**
τ_g	0.2	0.377±0.009	1.464±0.184	0.285±0.001	7.977±0.110
	0.8	**0.364±0.002**	1.421±0.259	0.285±0.001	7.962±0.057
	1.0	0.371±0.009	**1.292±0.166**	0.283±0.003	7.843±0.046
	2.0	0.373±0.011	1.414±0.038	**0.279±0.001**	**7.717±0.043**

F Full Benchmark Results

This section reports the complete horizon-wise results supporting the averaged comparison in the main paper. For each dataset and horizon, Table F.1 reports the mean and standard error over three independent runs.

HyperDiff achieves the lowest CRPS mean for every dataset and horizon in this comparison. The result indicates that the high-order structure-driven diffusion design improves distributional accuracy at each forecasting length, not only after averaging. Standard errors are generally small on the ETT, Traffic, and Electricity benchmarks. The differences are more pronounced in difficult long-horizon settings such as Exchange Rate and ETTm2, where uncertainty grows more rapidly and baselines degrade more strongly.

HyperDiff also achieves the lowest QICE mean in nearly all horizon-wise comparisons and after averaging across horizons. Some QICE standard errors are larger in challenging long-horizon settings because interval calibration is more sensitive when the future distribution is more uncertain. HyperDiff nevertheless retains the lowest mean calibration error across datasets in the reported comparison.

G Additional Efficiency Analysis

We further examine the computational efficiency of HyperDiff to assess whether its probabilistic forecasting gains come at disproportionate cost. The analysis considers accuracy, inference cost, and model size on representative datasets, together with parameter counts and MACs across all seven benchmarks.

Computational efficiency. On ETTh1 and Exchange Rate, HyperDiff achieves lower CRPS than the compared methods while maintaining moderate inference cost and model size, as shown in Figure G.1. Compared with lightweight baselines such as NsDiff and TMDM, HyperDiff requires additional computation for dynamic high-order structure learning and structure-adaptive scheduling. This moderate overhead is accompanied by improved probabilistic forecasting accuracy. Compared with larger generative baselines such as DiffusionTS, FlowTS, TimeDiff, and KOONPRO, HyperDiff uses substantially fewer parameters while achieving lower CRPS on both representative datasets. These results suggest that the forecasting gains arise from targeted structural and heteroscedastic modeling rather than simply increased model capacity.

Parameter and computational complexity. Table G.1 reports parameter counts and MACs for all compared methods across the seven datasets. HyperDiff is larger than lightweight baselines

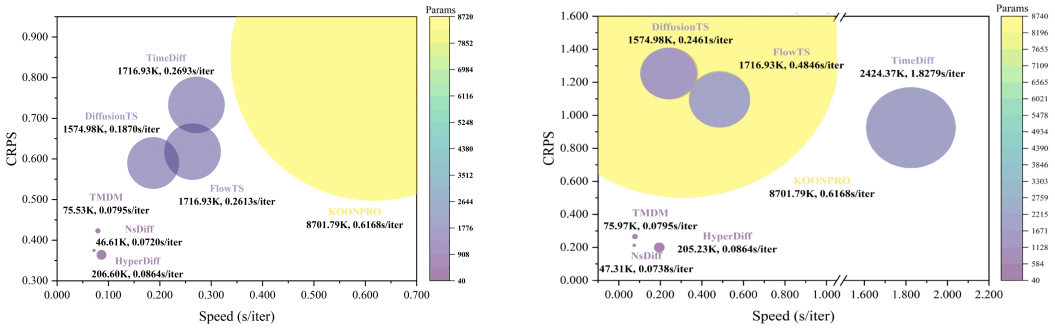
Table F.1: Full horizon-wise CRPS and QICE (%) corresponding to the main-paper results table. Values are mean ± standard error over three runs; lower is better; lowest values are bold.

Metric	Dataset	Horizon	HyperDiff (Ours)	NsDiff (2025)	KOONPRO (2025)	TMDM (2024)	DiffusionTS (2024)	FlowTS (2024)	TimeDiff (2023)
CRPS↓	ETTh1	96	**0.364±0.005**	0.375±0.009	0.850±0.004	0.423±0.015	0.590±0.003	0.618±0.053	0.733±0.006
		192	**0.382±0.004**	0.418±0.018	0.781±0.020	0.474±0.002	0.604±0.007	0.631±0.022	0.725±0.019
		336	**0.420±0.007**	0.423±0.014	0.716±0.001	0.443±0.001	0.606±0.002	0.639±0.011	0.710±0.030
		720	**0.385±0.012**	0.456±0.030	0.724±0.004	0.523±0.037	0.661±0.003	0.631±0.002	1.113±0.007
	ETTh2	96	**0.294±0.011**	0.300±0.002	1.625±0.007	0.330±0.012	1.133±0.045	1.253±0.163	1.125±0.024
		192	**0.368±0.005**	0.376±0.015	1.665±0.016	0.377±0.009	1.148±0.033	1.329±0.073	1.126±0.013
		336	**0.402±0.008**	0.417±0.014	1.700±0.039	0.460±0.005	1.226±0.040	1.312±0.023	0.714±0.009
		720	**0.332±0.021**	0.356±0.014	1.700±0.039	0.408±0.012	1.143±0.005	1.120±0.030	0.727±0.019
	ETTm1	96	**0.296±0.001**	0.331±0.014	0.730±0.001	0.304±0.004	0.509±0.013	0.488±0.007	0.726±0.037
		192	**0.333±0.003**	0.346±0.011	0.784±0.024	0.346±0.007	0.536±0.017	0.579±0.076	0.726±0.037
		336	**0.346±0.004**	0.350±0.007	0.786±0.006	0.374±0.016	0.565±0.003	0.794±0.076	0.711±0.014
		720	**0.391±0.001**	0.418±0.019	0.766±0.006	0.420±0.011	0.649±0.005	0.673±0.008	0.791±0.248
	ETTm2	96	**0.230±0.001**	0.237±0.004	1.582±0.074	0.233±0.005	0.996±0.035	0.966±0.034	1.155±0.073
		192	**0.244±0.002**	0.270±0.005	1.609±0.016	0.280±0.010	1.056±0.047	1.287±0.102	1.136±0.026
		336	**0.316±0.003**	0.339±0.008	1.671±0.034	0.374±0.033	1.099±0.020	1.147±0.237	0.841±0.208
		720	**0.385±0.010**	0.517±0.052	1.524±0.099	0.450±0.024	1.138±0.039	1.201±0.077	1.199±0.028
	Exchange Rate	96	**0.200±0.004**	0.213±0.006	1.371±0.057	0.266±0.062	1.253±0.007	1.098±0.064	0.930±0.013
		192	**0.265±0.010**	0.278±0.005	1.348±0.008	0.378±0.030	1.259±0.016	1.153±0.034	1.979±1.085
		336	**0.359±0.013**	0.385±0.016	1.346±0.009	0.388±0.040	1.274±0.020	1.218±0.027	1.124±0.036
		720	**0.544±0.021**	0.766±0.051	1.731±0.052	0.848±0.276	1.562±0.006	1.597±0.005	0.782±0.000
	Traffic	96	**0.370±0.004**	0.380±0.014	0.786±0.000	0.558±0.003	0.711±0.002	0.641±0.005	0.767±0.000
		192	**0.371±0.006**	0.376±0.004	0.763±0.000	0.562±0.000	0.709±0.002	0.625±0.005	0.778±0.002
		336	**0.369±0.000**	0.385±0.002	0.769±0.000	0.566±0.004	0.718±0.003	0.621±0.003	0.779±0.001
		720	**0.371±0.003**	0.374±0.002	0.779±0.000	0.565±0.000	0.735±0.000	0.653±0.001	0.766±0.001
	Electricity	96	**0.288±0.001**	0.295±0.002	0.795±0.000	0.494±0.000	0.648±0.000	0.409±0.016	0.789±0.003
		192	**0.291±0.001**	0.306±0.005	0.810±0.000	0.495±0.000	0.649±0.004	0.438±0.020	0.767±0.002
		336	**0.293±0.000**	0.303±0.002	0.796±0.000	0.495±0.001	0.649±0.028	0.448±0.016	0.767±0.002
		720	**0.306±0.001**	0.312±0.001	0.802±0.000	0.499±0.001	0.658±0.001	0.633±0.011	0.781±0.003
QICE (%)↓	ETTh1	96	**1.216±0.171**	1.596±0.078	14.415±0.014	2.771±0.689	4.078±0.207	9.070±0.681	15.164±0.052
		192	**1.307±0.789**	1.589±0.178	14.357±0.045	3.296±0.177	5.766±0.221	7.742±0.613	15.178±0.049
		336	**1.240±0.127**	1.306±0.327	14.351±0.027	3.527±0.407	6.872±0.421	7.404±0.398	14.505±0.367
		720	**1.459±0.348**	3.617±0.905	14.451±0.016	4.457±0.602	9.449±0.261	4.369±0.138	14.703±0.080
	ETTh2	96	**1.711±0.159**	2.066±0.064	16.417±0.121	3.395±0.354	8.553±0.094	11.710±0.429	15.201±0.254
		192	**2.505±0.278**	2.565±0.127	16.946±0.259	3.552±0.424	9.196±0.569	11.258±0.569	15.879±0.343
		336	**2.549±0.203**	3.500±0.096	17.027±0.254	4.168±0.293	10.722±0.571	11.227±0.589	15.070±0.030
		720	**2.204±0.133**	2.456±0.500	17.025±0.253	2.216±0.214	12.615±0.085	8.813±0.057	14.982±0.166
	ETTm1	96	**1.961±0.365**	2.343±0.012	14.412±0.045	2.762±0.089	5.688±0.390	9.524±0.268	15.091±0.026
		192	**1.540±0.272**	2.189±0.123	14.342±0.036	3.087±0.412	5.105±0.496	10.656±1.161	15.087±0.031
		336	**1.633±0.198**	1.754±0.142	14.332±0.025	2.399±0.227	4.155±0.166	12.912±1.307	15.125±0.022
		720	**1.696±0.419**	2.044±0.296	14.371±0.019	3.255±0.863	3.933±0.120	9.838±0.327	14.274±0.819
	ETTm2	96	**1.721±0.533**	2.682±0.511	15.339±0.272	1.912±0.155	11.124±0.363	13.149±0.645	15.314±0.206
		192	**1.872±0.534**	2.806±0.301	15.777±0.180	2.707±0.597	10.831±0.508	12.930±0.738	15.075±0.221
		336	**1.797±0.078**	3.063±0.107	16.338±0.106	1.845±0.481	9.232±0.129	13.226±0.592	14.609±0.559
		720	**3.362±0.323**	5.950±0.838	15.676±0.692	3.365±0.841	8.648±0.113	10.983±0.294	16.015±0.178
	Exchange Rate	96	**4.082±0.385**	4.485±0.421	15.295±0.036	5.083±0.368	9.199±0.090	10.945±0.392	14.754±0.493
		192	**3.787±0.265**	5.855±1.274	15.405±0.071	5.955±0.875	8.963±0.034	10.563±0.290	14.997±0.235
		336	**4.050±0.605**	5.841±0.318	15.535±0.048	6.692±1.030	9.066±0.009	10.179±0.098	15.476±0.156
		720	**5.001±0.808**	9.143±0.810	15.810±0.052	6.918±1.353	12.940±0.023	13.648±0.118	15.126±0.002
	Traffic	96	**2.550±0.058**	8.022±0.480	14.992±0.001	10.668±0.118	13.220±0.006	6.747±0.017	15.110±0.002
		192	**2.551±0.116**	8.397±0.152	14.971±0.000	10.659±0.095	13.276±0.020	5.593±0.349	15.106±0.005
		336	**2.044±0.008**	8.294±0.054	14.978±0.000	10.573±0.027	13.517±0.048	4.726±0.183	15.127±0.002
		720	**2.252±0.156**	8.134±0.138	14.983±0.000	10.488±0.071	13.984±0.013	3.318±0.081	15.415±0.019
	Electricity	96	**6.986±0.049**	7.939±0.082	15.260±0.001	11.102±0.043	14.795±0.407	8.999±0.183	15.398±0.007
		192	**5.619±0.166**	7.363±0.133	15.258±0.001	10.928±0.071	13.460±0.550	9.128±0.269	15.397±0.009
		336	**3.666±0.073**	7.298±0.181	15.265±0.001	10.949±0.054	14.648±0.090	9.333±0.342	15.368±0.017
		720	**3.277±0.007**	7.006±0.073	15.250±0.001	10.702±0.048	14.262±0.052	8.531±0.898	14.754±0.493

such as NsDiff and TMDM, reflecting the additional structural learning and adaptive scheduling components, but remains substantially smaller than several larger generative models. The overall results support a favorable balance between probabilistic forecasting performance and computational complexity.

H Additional Residual Dependence Analysis

We further analyze whether changes in cross-variable residual dependence are associated with the learned dynamic high-order structure. This analysis is intended as a diagnostic of structure-aligned model behavior rather than as direct evidence of improved joint-distribution quality, which is evaluated separately by Energy Score and Variogram Score.



(a) ETTh1. (b) Exchange Rate.

Figure G.1: Accuracy and computational efficiency across methods. (a) ETTh1 and (b) Exchange Rate, comparing forecasting accuracy with computational cost and model size.

Table G.1: Parameter counts and MACs across datasets. Params are in K and MACs in M.

Metric	Model	ETTh1	ETTh2	ETTm1	ETTm2	Exch.	Traffic	Elec.
Params (K)↓	HyperDiff (Ours)	204.04	206.39	206.42	206.42	205.23	1284.24	604.87
	NsDiff (2025)	46.60	46.60	46.63	46.63	47.31	677.56	278.37
	KOONPRO (2025)	8701.79	8701.79	8701.79	8701.79	8722.32	26254.94	15148.21
	TMDM (2024)	75.52	75.52	75.55	75.55	75.97	486.75	226.59
	DiffusionTS (2024)	1576.97	1574.97	1574.97	1574.97	1576.97	3282.41	2202.03
	FlowTS (2024)	1716.93	1716.93	1716.93	1716.93	1719.18	3643.24	2424.37
	TimeDiff (2023)	2424.37	3103.81	3103.81	3103.81	3123.47	19911.40	9276.42
MACs (M)↓	HyperDiff (Ours)	142.47	146.06	146.06	146.06	145.18	3111.14	1125.77
	NsDiff (2025)	118.90	118.90	119.07	119.07	121.21	2240.99	898.47
	KOONPRO (2025)	2395.58	2395.58	2395.58	2395.58	2470.41	66378.58	25893.43
	TMDM (2024)	208.70	208.70	208.87	208.87	211.26	2540.91	1065.44
	DiffusionTS (2024)	6376.61	6376.61	6376.61	6376.61	6380.52	9717.76	7603.65
	FlowTS (2024)	6391.75	6391.75	6391.75	6391.75	6397.82	11582.00	8297.88
	TimeDiff (2023)	8297.88	8884.56	8884.56	8884.56	8896.58	19159.66	12658.11

Residual-dependence definition. For each test window and future position, let $\mu_{p,i}$ and $\sigma_{p,i}$ denote the empirical mean and standard deviation of the predictive samples for variable i. We define the standardized forecast residual as

$$z_{p,i} = \frac{y_{p,i} - \mu_{p,i}}{\sigma_{p,i} + \varepsilon_{\text{num}}} \tag{H.1}$$

The residual-dependence matrix R is obtained by computing Pearson correlations between standardized residuals after aggregating all test windows and future positions.

For the structural analysis, the final incidence matrix is first normalized over hyperedges for each variable:

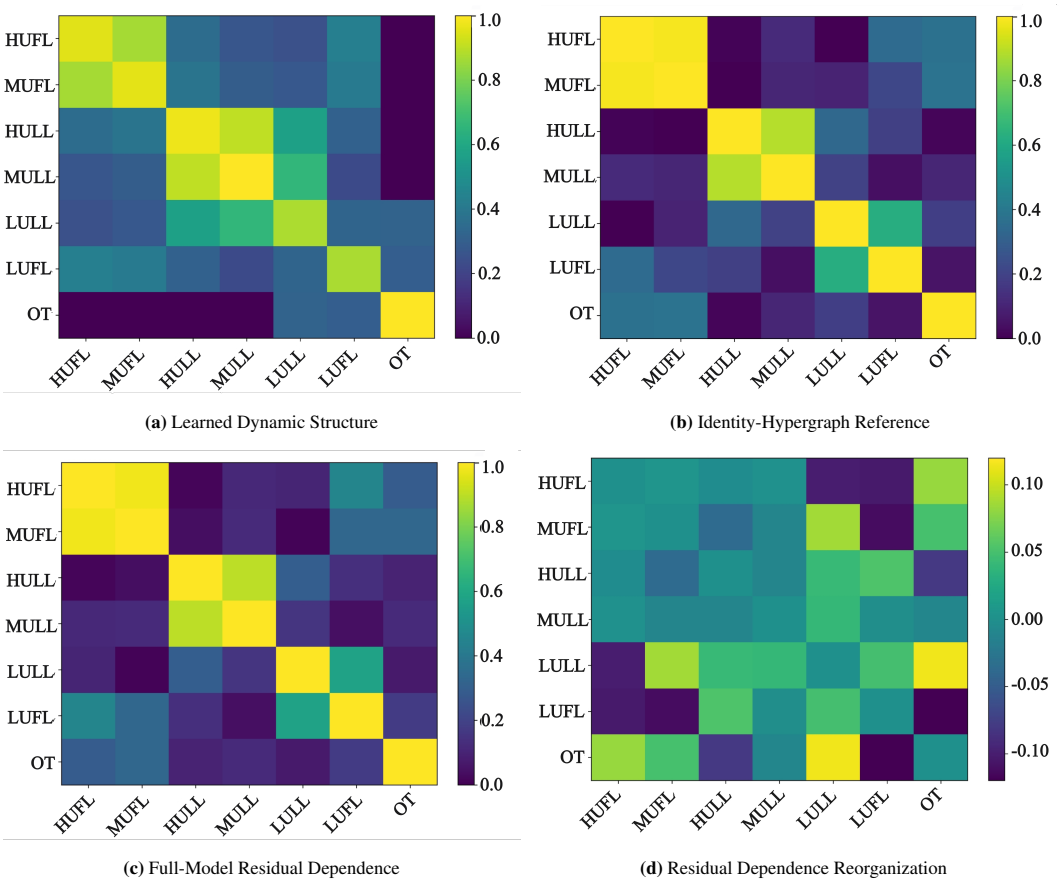
$$\bar{h}_{i,e} = \frac{h_{i,e}}{\sum_{e'} h_{i,e'} + \varepsilon_{\text{num}}} \tag{H.2}$$

The structural co-membership score for variables i and j is then

$$S_{ij} = \mathbb{E}_{\text{test}} \left[\sum_{e} \bar{h}_{i,e} \bar{h}_{j,e} \right] \tag{H.3}$$

where the expectation is taken over test windows. Larger S_{ij} indicates that the two variables more frequently receive shared high-order structural context.

Reference model and dependence change. The reference is a separately trained identity-hypergraph variant in which the incidence matrix is fixed to the identity matrix, so each variable is assigned to its own hyperedge and no learned cross-variable hypergraph membership is used. Let



(a) Learned Dynamic Structure

(b) Identity-Hypergraph Reference

(c) Full-Model Residual Dependence

(d) Residual Dependence Reorganization

Figure H.1: Residual-dependence analysis on ETTh1 with prediction horizon 96. (a) Learned dynamic structure, (b) identity-hypergraph reference, (c) full-model residual dependence, and (d) residual-dependence reorganization.

R^{full} and R^{ref} denote the residual-dependence matrices of HyperDiff and this reference, respectively. We define the pairwise dependence change as

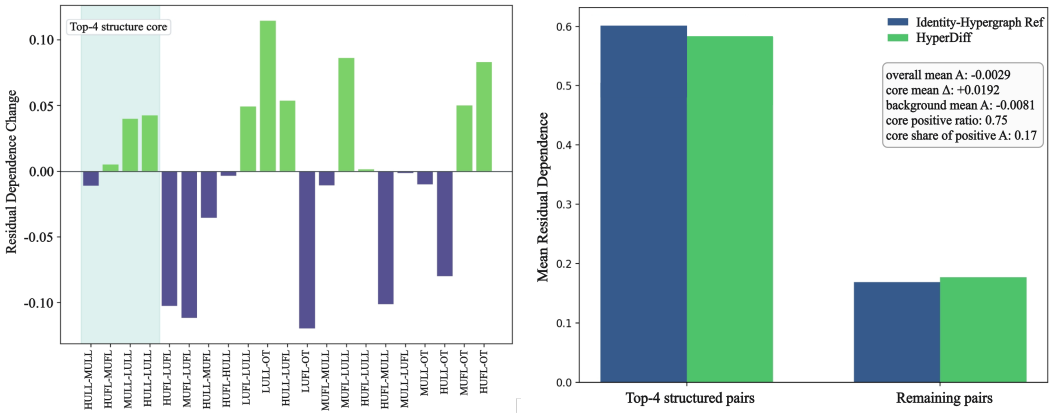
$$\Delta_{ij} = \left|R^{\text{ref}}_{ij}\right| - \left|R^{\text{full}}_{ij}\right| \tag{H.4}$$

Positive Δ_{ij} therefore denotes weaker absolute standardized residual dependence under HyperDiff relative to the identity-hypergraph reference.

Dynamic structure and residual dependence. The learned high-order structure exhibits nonuniform co-membership patterns, indicating that variables do not receive uniformly shared structural context. As shown in Figure H.1, the identity-hypergraph reference retains stronger absolute residual dependence for several variable pairs, whereas HyperDiff exhibits nonuniform changes across pairs. Some dependencies are weakened while others receive smaller or mixed adjustments, which is consistent with structure-dependent rather than globally uniform reorganization.

Structure-aligned dependence reorganization. We rank all variable pairs by the structural co-membership score S_{ij} in descending order. Following the analysis configuration, the top 20% of pairs are defined as the *structural core*, while the remaining pairs form the *background* set.

Figure H.2(a) reports Δ_{ij} after ranking pairs by S_{ij}, allowing the magnitude and direction of residual-dependence changes to be compared with structural relevance. Figure H.2(b) further compares the mean absolute residual dependence of the structural-core and background pairs under the identity-hypergraph reference and HyperDiff. The larger reductions observed among highly ranked pairs indicate that dependence changes are more strongly associated with structurally relevant relations. This result should be interpreted as a diagnostic association, rather than as evidence that weaker residual correlation is inherently better.



(a) Structure-ranked Change **(b)** Core vs. Background

Figure H.2: Structure-aligned residual-dependence analysis on ETTh1 with prediction horizon 96. (a) Structure-ranked dependence change and (b) comparison between structural-core and background pairs.

I Robustness Analysis under Temporal Distribution Shift

We further evaluate HyperDiff under three evaluation-time temporal conditions—volatility shift, regime change, and anomalous windows—using ETTh1 with prediction horizon 96. The corresponding grouping statistics are computed from the observed test windows only for post-hoc evaluation; they are not provided to the forecasting model and do not affect prediction generation.

Shift definitions. For a future window $Y \in \mathbb{R}^{P\times N}$, the volatility score is defined as the root-mean-square first-order change across future positions and variables:

$$s_{\text{vol}} = \sqrt{\frac{1}{(P-1)N}\sum_{p=2}^{P}\sum_{i=1}^{N}(Y_{p,i} - Y_{p-1,i})^2} \tag{I.1}$$

To characterize a regime change at the history–future boundary, we measure the normalized jump between the last historical observation and the first future observation:

$$s_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_{1,i} - X_{L,i}|}{\text{Std}(X_{1:L,i}) + \varepsilon_{\text{num}}} \tag{I.2}$$

For anomalous windows, we use a robust deviation score. Let $m_i = \text{median}_p(Y_{p,i})$ and $\text{MAD}_i = \text{median}_p\,|Y_{p,i} - m_i|$. The anomaly score is

$$s_{\text{ano}} = \max_{p,i}\frac{|Y_{p,i} - m_i|}{1.4826\ \text{MAD}_i + \varepsilon_{\text{num}}} \tag{I.3}$$

For each statistic, test windows are ranked independently. The bottom 30% and top 30% form the two comparison groups: low versus high volatility, stable versus regime-shift, and normal versus anomalous windows, respectively. The middle 40% are excluded from the corresponding group comparison.

Across the three evaluation conditions, HyperDiff maintains relatively low CRPS and comparatively stable QICE despite local fluctuations, as shown in Figure I.1. These observations are consistent with stable probabilistic forecasting behavior across the evaluated temporal conditions.

This behavior is consistent with the window-conditioned design of HyperDiff. Dynamic high-order structure learning adapts the cross-variable organization to the current historical window, while history-derived volatility and structure-adaptive scheduling provide state-dependent conditioning for stochastic generation. Corresponding robustness curves for the available baseline methods are reported in Figures I.2, I.3, and I.4.

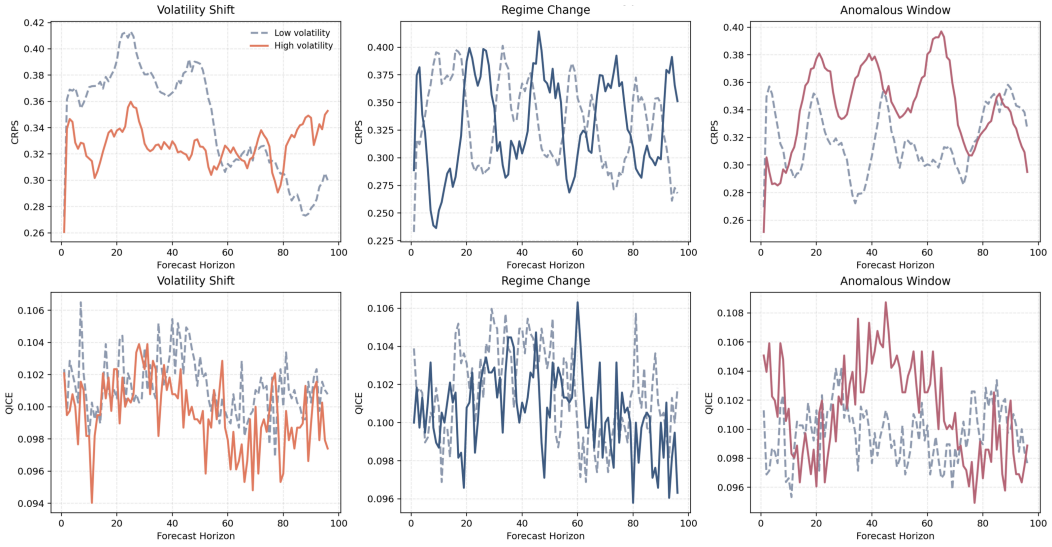


Figure I.1: Temporal-shift robustness of HyperDiff on ETTh1 with prediction horizon 96. CRPS and QICE are reported under volatility shift, regime change, and anomalous-window scenarios.

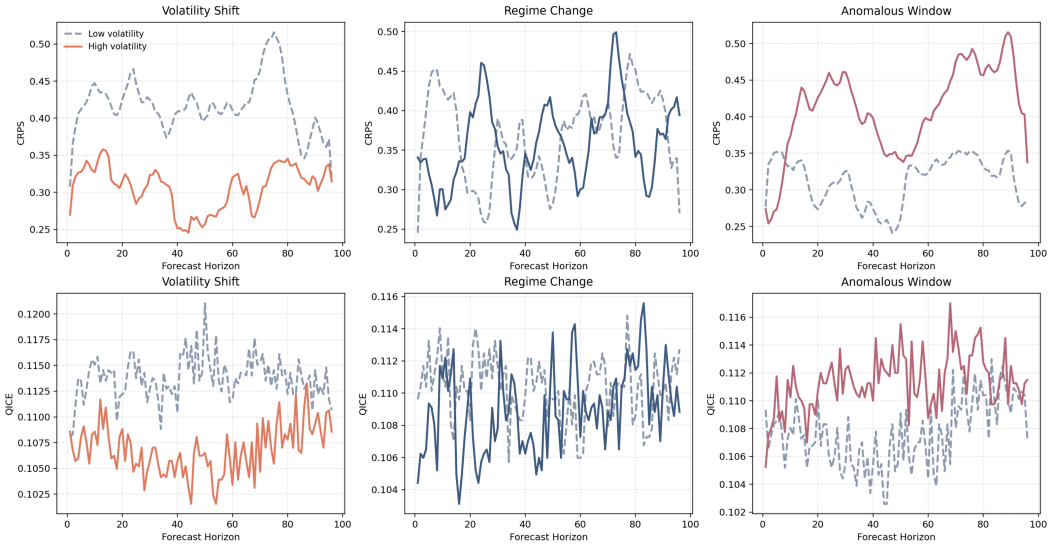


Figure I.2: Temporal-shift robustness of TMDM on ETTh1 with prediction horizon 96. CRPS and QICE are reported under three shift scenarios.

For comparison, the corresponding temporal-shift robustness curves for the available baseline methods are shown below.

J Qualitative Forecasting Visualization

In addition to the quantitative results, we provide a qualitative comparison on one sample. Figure J.1 shows the median forecasts and prediction intervals produced by HyperDiff and the competing methods. This case study complements the main-paper results table and Table F.1 by showing how the numerical differences appear in forecast shapes and uncertainty bands.

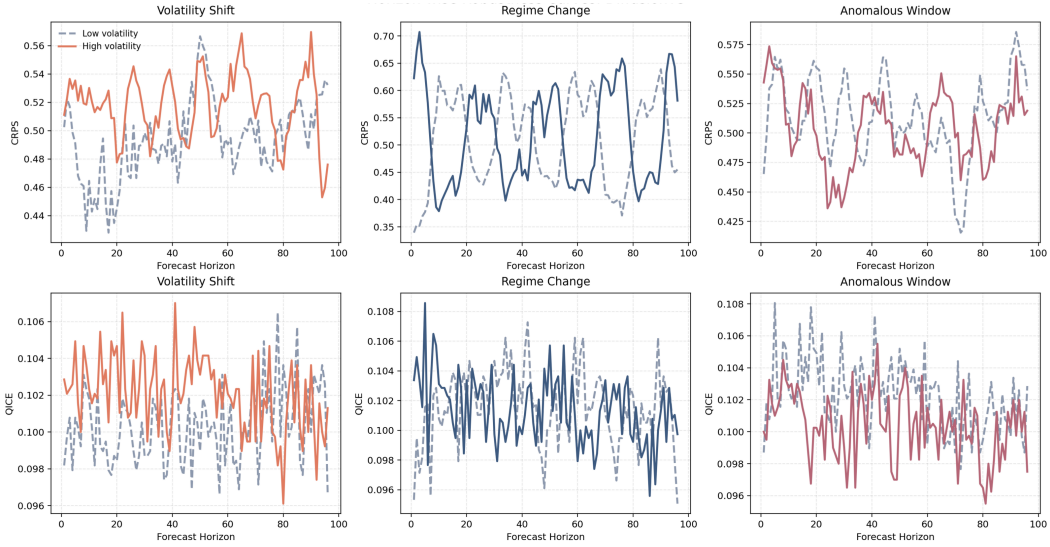


Figure I.3: Temporal-shift robustness of DiffusionTS on ETTh1 with prediction horizon 96. CRPS and QICE are reported under three shift scenarios.

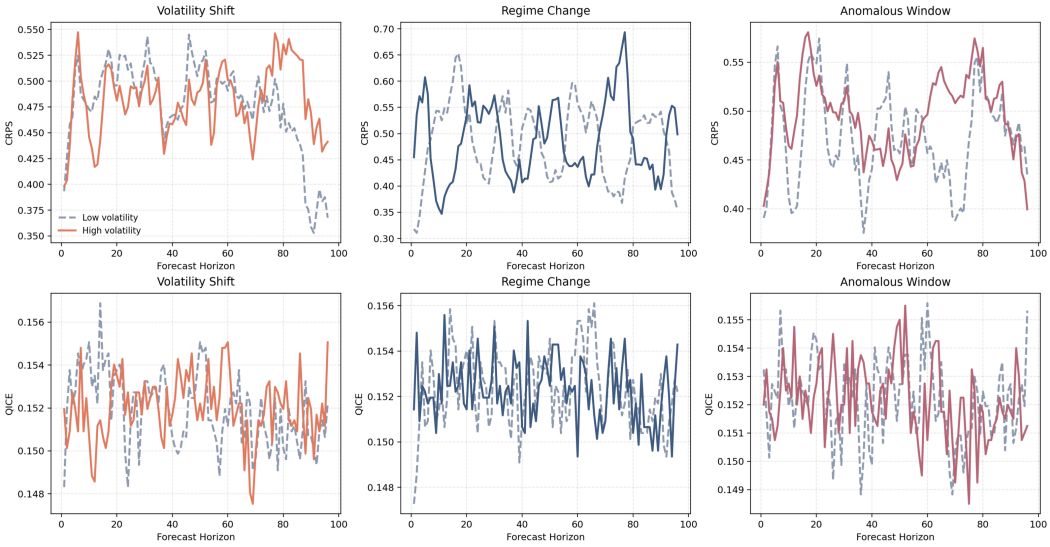


Figure I.4: Temporal-shift robustness of TimeDiff on ETTh1 with prediction horizon 96. CRPS and QICE are reported under three shift scenarios.

In this example, HyperDiff produces a median forecast closer to the ground-truth trajectory. Its prediction intervals are also more compact while covering most observed values. NsDiff and TMDM capture the overall trend, but their intervals are wider and their local biases are more pronounced in the displayed window. FlowTS and DiffusionTS show larger forecast oscillations, whereas KOONPRO and TimeDiff exhibit more evident distributional shifts. These patterns are consistent with the quantitative results, but they remain qualitative evidence from one representative case rather than a standalone performance claim.

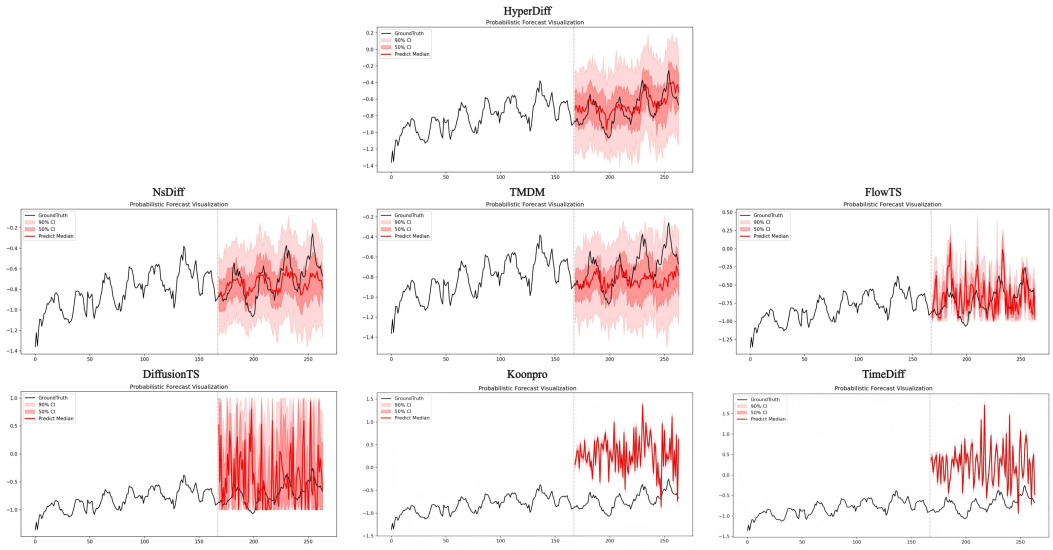


Figure J.1: Qualitative probabilistic forecasting on ETTh1 with prediction horizon 96. The figure compares median forecasts and prediction intervals across methods.