CTD-MTSI: CNN–TRANSFORMER DIFFUSION FOR PROBABILISTIC MULTIVARIATE TIME SERIES IMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent diffusion-based multivariate time series imputation (MTSI) approaches provide strong probabilistic modeling but rely on attention mechanisms for learning both temporal dynamics and cross-variable dependencies. This work introduces CTD-MTSI, a conditional CNN–transformer diffusion architecture that decouples temporal modeling and cross-variable interaction. Depthwise temporal convolutions explicitly capture local dynamics, while attention inter-variable dependencies. Contextual information is injected through mask-aware embeddings and feature-wise modulation during denoising. Experiments on benchmark datasets show that CTD-MTSI consistently outperforms the state-of-the-art diffusion baseline CSDI in both reconstruction accuracy and uncertainty quantification. These results highlight the effectiveness of structured convolutional inductive bias within conditional diffusion models for time series imputation.

Track: Research

1 INTRODUCTION AND RELATED WORK

Multivariate time series (MTS) data appear in many real-world systems such as healthcare monitoring, finance, traffic, and climate science (Yi et al., 2016; Tan et al., 2013; Nelwamondo et al., 2007; Hudak et al., 2008; van Buuren & Groothuis-Oudshoorn, 2011). Each variable (or sensor) follows its own temporal dynamics, while exhibiting inter-variable correlations. Modeling these temporal and cross-variable dependencies is critical for forecasting, decision making and understanding system dynamics. Multivariate time series imputation (MTSI) aims to recover missing values by exploiting both temporal continuity and relationships among variables.

Early deep learning approaches for MTSI focused on deterministic point estimation. Recurrent models propagate imputed values through bidirectional recurrent structures, capturing temporal dynamics (Che et al., 2016; Cao et al., 2018). Attention-based architectures further improved global dependency modeling by leveraging self-attention and have rapidly gained dominance in time series tasks due to their scalability and strong long-range modeling capacity (Suo et al., 2020; Bansal et al., 2021; Du et al., 2023). However, these methods produce single deterministic reconstructions, which may underestimate uncertainty. To address uncertainty estimation, variational approaches incorporate probabilistic inference (Mulyadi et al., 2022). Gaussian process–based methods (Bonilla et al., 2007; Fortuin et al., 2020) provide principled uncertainty modeling but scale poorly with high-dimensional data. Although these methods improve uncertainty calibration, modeling highly nonlinear dependencies remains challenging.

Diffusion models have recently become the state-of-the-art for probabilistic MTSI, as they formulate imputation as a reverse denoising process conditioned on observed entries. By learning the conditional score function, these models generate multiple plausible imputations and achieve strong performance. Architecturally, most of diffusion-based methods build upon a DiffWave-style residual backbone (Kong et al., 2021), adapted by introducing separate attention mechanisms applied sequentially along temporal and feature dimensions within the same residual diffusion block. However, conditioning is typically implemented via token concatenation, requiring attention layers to implicitly separate observed context from noisy targets. Also, noisy temporal representations may be propagated across variables before sufficient local denoising occurs. Convolutional architectures

have received comparatively less recognition than attention-based models in MTSI due to perceived inferior modeling capability. Yet, convolution provides strong local inductive bias, motivating this work, which investigates whether convolution can be more effectively integrated into probabilistic imputation.

To this end, CTD-MTSI introduces a conditional CNN–transformer diffusion architecture for probabilistic MTSI. Contextual information (observed entries) is injected directly into intermediate representations via mask-aware patch embeddings and a feature-wise linear modulation mechanism. A hierarchical backbone combines depthwise temporal convolutions with variable attention, providing multi-scale temporal modeling. Such structural decoupling simplifies optimization compared to jointly modeling temporal and cross-variable dependencies within the same attention-based residual block.

2 METHODOLOGY

We consider the problem of probabilistic multivariate time series imputation. Let $\mathbf{X}_0 \in \mathbb{R}^{V \times L}$ denote a multivariate time series with V variables observed over L time steps. Due to missing entries, the full signal is only partially observed. We define a binary observation mask $\mathbf{M} \in \{0,1\}^{V \times L}$, where $\mathbf{M}_{v,\ell} = 1$ indicates that the value $\mathbf{X}_{v,\ell}$ is observed and $\mathbf{M}_{v,\ell} = 0$ otherwise. The partially observed data is then given by the element-wise product $\mathbf{X}_0^{\text{ob}} = \mathbf{M} \odot \mathbf{X}$, where $\odot$ denotes Hadamard (element-wise) multiplication.

2.1 CONDITIONAL DENOISING DIFFUSION PROBABILISTIC MODEL

To perform probabilistic imputation, we model the conditional distribution $p(\mathbf{X}_0^{\text{mi}} \mid \mathbf{X}_0^{\text{ob}}, \mathbf{M})$, using a conditional denoising diffusion probabilistic model (Ho et al., 2020), which allows to generate multiple plausible imputations and quantify uncertainty. The diffusion process is applied only to the missing part, while $\mathbf{X}_0^{\text{ob}}$ remains fixed and serves as contextual information throughout the reverse process.

Forward process. We define a Markov chain that gradually transforms the missing values into Gaussian noise:

$$q(\mathbf{X}_t^{\text{mi}} \mid \mathbf{X}_{t-1}^{\text{mi}}) = \mathcal{N}\big(\mathbf{X}_t^{\text{mi}}; \sqrt{\alpha_t}\mathbf{X}_{t-1}^{\text{mi}}, (1-\alpha_t)\mathbf{I}\big)\,, \tag{1}$$

where t is the noise level, $\{\alpha_t\}_{t=1}^T$ is a predefined variance schedule and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$. Then $\mathbf{X}_t^{\text{mi}}$ can be expressed as $\mathbf{X}_t^{\text{mi}} = \sqrt{\bar{\alpha}_t}\mathbf{X}_0^{\text{mi}} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus, at an arbitrary timestep t, the noisy missing variables $\mathbf{X}_t^{\text{mi}}$ can be obtained directly from the clean $\mathbf{X}_0^{\text{mi}}$ by injecting Gaussian noise.

Conditional reverse process. The generative model is defined as a reverse-time Markov chain conditioned on the observed context,

$$p_\theta(\mathbf{X}_{0:T}^{\text{mi}} \mid \mathbf{X}_0^{\text{ob}}, \mathbf{M}) = p(\mathbf{X}_T^{\text{mi}}) \prod_{t=1}^{T} p_\theta\big(\mathbf{X}_{t-1}^{\text{mi}} \mid \mathbf{X}_t^{\text{mi}}, \mathbf{X}_0^{\text{ob}}, \mathbf{M}\big)\,, \quad \mathbf{X}_T^{\text{mi}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $\mathbf{X}_T^{\text{mi}}$ is initialized as pure Gaussian noise and progressively refined toward a plausible imputation. Each reverse transition is modeled as a Gaussian distribution, defined as

$$p_\theta\big(\mathbf{X}_{t-1}^{\text{mi}} \mid \mathbf{X}_t^{\text{mi}}, \mathbf{X}_0^{\text{ob}}, \mathbf{M}\big) = \mathcal{N}\big(\mathbf{X}_{t-1}^{\text{mi}}; \boldsymbol{\mu}_\theta(\mathbf{X}_t^{\text{mi}}, t \mid \mathbf{X}_0^{\text{ob}}, \mathbf{M}), \sigma_t^2\mathbf{I}\big)\,. \tag{3}$$

Following the standard DDPM parameterization, the mean is expressed through a conditional noise predictor $\boldsymbol{\epsilon}_\theta(\mathbf{X}_t^{\text{mi}}, t \mid \mathbf{X}_0^{\text{ob}}, \mathbf{M})$, which corresponds to predicting and removing the injected noise at each diffusion step. The model is trained using the standard denoising objective. At inference time, we iteratively sample from $t = T$ down to $t = 0$, obtaining multiple plausible imputations $\mathbf{X}_0^{\text{mi}}$ that are consistent with $\mathbf{X}_0^{\text{ob}}$ and respect the observed mask. Full derivations and sampling details are provided in Appendix A.2.4.

2.2 NOISE PREDICTOR ARCHITECTURE

We parameterize the conditional noise predictor $\boldsymbol{\epsilon}_\theta(\mathbf{X}_t^{\text{mi}}, t \mid \mathbf{X}_0^{\text{ob}}, \mathbf{M})$ using a CNN-transformer backbone. The design combines (i) convolutional patch embeddings and depthwise temporal mixing

with (ii) multi-head attention blocks to capture cross-variable dependencies. A schematic overview of the full backbone is provided in Appendix A.4.

Patch and mask-aware embedding. Both the observed context and the noisy missing target are mapped into a shared latent feature space using a convolutional patch stem. The stem consists of a 1D convolution applied along the temporal dimension with kernel size p (patch size) and stride s (patch stride). This operation aggregates short temporal windows into latent tokens while reducing the effective sequence length. To make the embedding mask-aware, the binary mask is concatenated as an additional input channel, allowing the network to distinguish observed from missing values. Let $\mathcal{E}(\cdot)$ denote the stem embedding function. The embedded representations are computed as $\mathbf{Z}^{\text{ob}} = \mathcal{E}(\mathbf{X}_0^{\text{ob}}, \mathbf{M})$ and $\mathbf{Z}_t^{\text{mi}} = \mathcal{E}(\mathbf{X}_t^{\text{mi}}, \mathbf{1} - \mathbf{M})$, where $\mathbf{Z}^{\text{ob}}, \mathbf{Z}_t^{\text{mi}} \in \mathbb{R}^{B \times V \times H \times T'}$, B denotes the batch size, V the number of variables, H the number of latent feature channels (embedding dimension), and $T' = \lfloor (L - p)/s \rfloor + 1$ the patchified temporal length.

Fuse conditioning (FiLM). To inject the observed context, a feature-wise linear modulation (FiLM) mechanism (Perez et al., 2018) is applied at the stem resolution. Per-variable and per-timestep modulation parameters are predicted from the context embedding $\mathbf{Z}^{\text{ob}}$, while a separate modulation term is derived from a timestep embedding of the diffusion step t. Specifically, $(\boldsymbol{\gamma}^c, \boldsymbol{\beta}^c) = \mathrm{PWConv}(\mathbf{Z}^{\text{ob}})$ and $(\boldsymbol{\gamma}^t, \boldsymbol{\beta}^t) = \mathrm{MLP}(\mathrm{Emb}(t))$, which are combined as $\boldsymbol{\gamma} = \boldsymbol{\gamma}^c + \boldsymbol{\gamma}^t$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^c + \boldsymbol{\beta}^t$. The fused representation is then obtained as

$$\mathbf{Z}_t = (1 + \boldsymbol{\gamma}) \odot \mathbf{Z}_t^{\text{mi}} + \boldsymbol{\beta}. \tag{4}$$

Here, PWConv denotes a 1×1 convolution applied per variable without pooling, and $(\boldsymbol{\gamma}^t, \boldsymbol{\beta}^t)$ are broadcast across variables and time. This modulation preserves the backbone feature structure while conditioning the denoiser without directly concatenating noisy and observed channels at the stem.

Hierarchical CNN–transformer modules. The fused representation $\mathbf{Z}_t$ is processed by a hierarchy of backbone levels, each composed of multiple CNN–transformer blocks operating at a fixed temporal resolution. Between levels, temporal downsampling is performed using a strided 1D convolution, progressively enlarging the effective receptive field and enabling multi-scale temporal modeling.

Within each CNN–transformer block, (i) variable attention and (ii) a Conv-FFN for local refinement are alternated sequentially. Variable attention captures cross-variable dependencies, while temporal context is encoded using depthwise temporal convolutions inside the $Q/K/V$ projections, following modern convolutional designs for efficient long-sequence modeling (Donghao & Xue, 2024; Park et al., 2026). For an intermediate representation $\mathbf{H} \in \mathbb{R}^{B \times V \times H \times T'}$, multi-head variable attention is computed as

$$\mathrm{Attn}(\mathbf{H}) = \mathrm{Proj}\left(\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{T'}} \right) \mathbf{V} \right), \qquad \mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B \times H \times V \times T'}, \tag{5}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are produced via shared depthwise temporal Conv1D mixers and Proj denotes a Conv 1×1 projection. Each block is further conditioned on the diffusion timestep through a sinusoidal timestep embedding followed by an MLP, which generates adaptive normalization (Peebles & Xie, 2023) scale and gating parameters that modulate the residual updates.

Prediction head. Finally, the latent features from the last backbone level are mapped to the noise prediction through a reconstruction head. A pixel-shuffle style temporal upsampling is first applied to restore the original temporal resolution, followed by a linear projection to the output channel corresponding to the predicted noise. Consistent with the backbone blocks, the prediction head is also conditioned on the diffusion timestep via adaptive layer normalization (AdaLN), whose scale and shift parameters are generated from the timestep embedding.

3 Experiments

3.1 Setup

The proposed method is evaluated on two commonly used multivariate time series benchmark datasets: ETTh1 and ETTh2 (Zhou et al., 2021). As a baseline, we compare against the state-of-the-art diffusion-based imputation model CSDI (Tashiro et al., 2021). Training follows a self-supervised masking scheme where 30% of observed entries are randomly masked and treated as

targets for reconstruction. To ensure a fair comparison, CSDI is trained under the same settings, including identical data splits and random seeds. Performance is measured using mean absolute error (MAE), root mean squared error (RMSE), and continuous ranked probability score (CRPS). Further implementation details are provided in Appendix A.

3.2 Results

Table 1 reports imputation performance under varying point wise missing ratios. Across both datasets, CTD-MTSI consistently matches or improves upon CSDI in all metrics under most missingness levels. Improvements are particularly consistent in the moderate regime, where temporal continuity alone becomes insufficient and cross-variable structure has stronger influence.

Table 1: Imputation performance under point missingness. Best results are marked in **red**.

Dataset	Missing	CTD-MTSI (Ours)			CSDI		
		MAE ↓	RMSE ↓	CRPS ↓	MAE ↓	RMSE ↓	CRPS ↓
ETTh1	0.1	**0.2310**	**0.3614**	**0.0361**	0.2383	0.3996	0.0373
	0.3	**0.3079**	**0.6205**	**0.0473**	0.3211	0.6347	0.0495
	0.5	**0.4292**	**0.9079**	**0.0665**	0.4499	0.9496	0.0694
	0.7	0.6974	1.5555	0.1092	**0.6712**	**1.4393**	**0.1037**
ETTh2	0.1	**0.3184**	**0.5465**	**0.0139**	0.3239	0.5749	0.0142
	0.3	**0.4166**	**0.7183**	**0.0183**	0.4248	0.7340	0.0187
	0.5	**0.5577**	**0.9429**	**0.0246**	0.5844	0.9904	0.0258
	0.7	**0.8882**	**1.4597**	**0.0394**	0.8977	1.5010	0.0396

While both methods rely on conditional diffusion, CTD-MTSI exhibits greater performance and robustness. The consistent gains under moderate missingness (30%–50%) regimes suggest that its decoupled temporal–variable modeling provides more stable information propagation during denoising. CRPS improvements further indicate improved probabilistic calibration. Lower CRPS values suggest that CTD-MTSI not only reduces reconstruction error but also produces closer predictive distributions.

Figure 1 provides a qualitative visualization under 70% missingness on ETTh2. The predicted median closely tracks ground truth trajectories, while the uncertainty bands widen appropriately in highly ambiguous regions. This behavior supports the quantitative findings and illustrates stable uncertainty estimation even under severe sparsity.

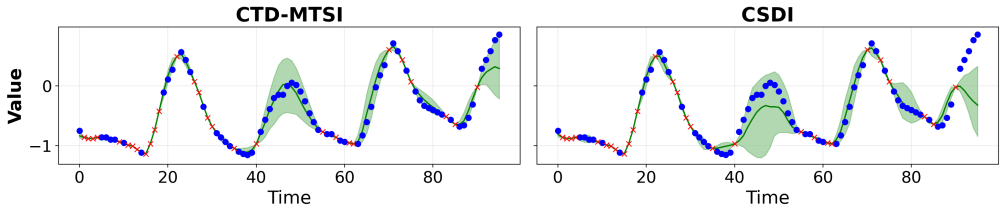

Figure 1: Visualization of probabilistic imputations results on ETTh2 with 70% random missingness. Median and 95% CIs shown in green; observed points in red; targets in blue.

4 Conclusion

This work introduces CTD-MTSI, a conditional CNN–transformer diffusion architecture for probabilistic multivariate time series imputation. By structurally decoupling temporal modeling from cross-variable interaction, the proposed design provides stable denoising behavior under varying levels of sparsity. Experimental results demonstrate competitive and often superior performance compared to the state-of-the-art diffusion-based baseline. These findings suggest that convolution, when carefully integrated within diffusion models, remains highly effective for time series imputation.

REFERENCES

Parikshit Bansal, Prathamesh Deshpande, and Sunita Sarawagi. Missing value imputation on multidimensional time series. *Proc. VLDB Endow.*, 14(11):2533–2545, 2021. doi: 10.14778/3476249.3476300.

Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Yitan Li, and Lei Li. Brits: bidirectional recurrent imputation for time series. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6776–6786, 2018.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values, 2016.

Luo Donghao and Wang Xue. ModernTCN: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.119619.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Raetsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1651–1661. PMLR, 2020.

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.

Andrew T. Hudak, Nicholas L. Crookston, Jeffrey S. Evans, David E. Hall, and Michael J. Falkowski. Nearest neighbor imputation of species-level, plot-scale forest structure attributes from lidar data. *Remote Sensing of Environment*, 112(5):2232–2245, 2008. doi: https://doi.org/10.1016/j.rse.2007.10.009.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

Ahmad Wisnu Mulyadi, Eunji Jun, and Heung-Il Suk. Uncertainty-aware variational-recurrent imputation network for clinical time series. *IEEE Transactions on Cybernetics*, 52(9):9684–9694, 2022. doi: 10.1109/TCYB.2021.3053599.

Fulufhelo V. Nelwamondo, Shakir Mohamed, and Tshilidzi Marwala. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pp. 1514–1521, 2007.

Dongik Park, Hyunwoo Ryu, Suahn Bae, Keondo Park, and Hyung-Sin Kim. T1: One-to-one channel-head binding for multivariate time-series imputation. In *The Fourteenth International Conference on Learning Representations*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

Qiuling Suo, Weida Zhong, Guangxu Xun, Jianhui Sun, Changyou Chen, and Aidong Zhang. Glima: Global and local time series imputation with multi-directional attention learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 798–807, 2020. doi: 10.1109/BigData50022.2020.9378408.

Huachun Tan, Guangdong Feng, Jianshuai Feng, Wuhong Wang, Yu-Jin Zhang, and Feng Li. A tensor-based method for missing traffic data completion. *Transportation Research Part C: Emerging Technologies*, 28:15–27, 2013. doi: https://doi.org/10.1016/j.trc.2012.12.007.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: conditional score-based diffusion models for probabilistic time series imputation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011. doi: 10.18637/jss.v045.i03.

Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. St-mvl: filling missing values in geo-sensory time series data. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2704–2710, 2016.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

A Implementation Details

A.1 Dataset Details

A.1.1 Dataset Description

Experiments are conducted on two multivariate time series datasets from the Electricity Transformer Temperature (ETT) benchmark (Zhou et al., 2021). The ETT datasets (ETTh1 and ETTh2) contain electricity transformer measurements recorded at an hourly frequency. Each dataset consists of 7 correlated variables representing oil temperature and load-related features. Table 2 summarizes the main statistics of the datasets.

Table 2: Dataset descriptions.

Dataset	Variables (V)	Train	Valid	Test	Frequency
ETTh1, ETTh2	7	8,545	2,785	2,785	Hourly

A.2 Experiment Details

A.2.1 Model Configuration

All experiments use a fixed input sequence length of 96. A one-dimensional convolutional embedding layer with kernel size 2 and stride 1 projects the input into a 128-dimensional channel space. The architecture consists of four CNN-transformer blocks organized into two hierarchical stages. The first stage contains two blocks using dual-scale depthwise convolutions with kernel sizes 71 and 5, followed by a downsampling layer with kernel size 2 and stride 2. The second stage contains two additional CNN-transformer blocks operating on the downsampled representation, with kernel sizes 31 and 5. This hierarchical design enables multi-scale temporal modeling, capturing both long-range and short-range dependencies. The feed-forward expansion ratio is set to 1.0. The configuration remains identical across both datasets.

A.2.2 Experimental Design

All experiments are conducted using three random seeds: 2025, 2026, and 2027. Training is performed on a NVIDIA RTX 6000 GPU. To evaluate generalization under different observability levels, masks are simulated following random point wise missingness. During training, a fixed masking ratio of 30% is applied using point wise random masking. The trained model is then evaluated under different test masking ratios (0.1, 0.3, 0.5, 0.7). This setup evaluates whether a model trained under a single missing ratio can generalize to different levels of observability without retraining.

A.2.3 Evaluation Metrics

Let V denote the number of variables and L the sequence length. Let $\mathcal{M}$ denote the set of artificially masked positions used for evaluation. Ground-truth values are denoted by $y_{v,\ell}$ and imputed values by $\hat{x}_{v,\ell}$, where $v \in \{1, \ldots, V\}$ and $\ell \in \{1, \ldots, L\}$. Mean absolute error (MAE), root mean squared error (RMSE), and continuous ranked probability score (CRPS) are employed. All metrics are computed only on artificially masked positions to ensure consistent and fair evaluation across models.

MAE The mean absolute error is defined as

$$\text{MAE} = \frac{1}{|\mathcal{M}|} \sum_{(v,\ell)\in\mathcal{M}} |\hat{x}_{v,\ell} - y_{v,\ell}| \,. \tag{6}$$

MAE measures the average magnitude of the imputation error. Each deviation contributes linearly to the final score, making MAE directly interpretable as the typical reconstruction error.

RMSE The root mean squared error is defined as

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{M}|} \sum_{(v,\ell)\in\mathcal{M}} (\hat{x}_{v,\ell} - y_{v,\ell})^2}. \tag{7}$$

RMSE computes the square root of the average squared error. By squaring deviations before averaging, larger errors are penalized more heavily. As a result, RMSE is more sensitive to outliers.

CRPS For probabilistic predictions with cumulative distribution function $F_{v,\ell}$ and ground truth $y_{v,\ell}$, CRPS is defined as

$$\text{CRPS} = \frac{1}{|\mathcal{M}|} \sum_{(v,\ell)\in\mathcal{M}} \int_{-\infty}^{\infty} \left(F_{v,\ell}(z) - \mathbb{I}(z \geq y_{v,\ell})\right)^2 dz. \tag{8}$$

CRPS evaluates the entire predictive distribution rather than a single point estimate. It measures how close the predicted cumulative distribution is to the empirical distribution concentrated at the ground-truth value, thus capturing both accuracy and uncertainty calibration.

A.2.4 Training Procedure

Following the conditional diffusion formulation described in Section 2.1. Let $X_0 \in \mathbb{R}^{V\times L}$ denote a multivariate time series with V variables and sequence length L. Define the binary observation mask $M \in \{0,1\}^{V\times L}$, where $M_{v,\ell} = 1$ indicates observed values and $M_{v,\ell} = 0$ indicates missing values. The observed and missing components are denoted as

$$X_0^{\text{ob}} = M \odot X_0, \qquad X_0^{\text{mi}} = (1 - M) \odot X_0.$$

During training, a self-supervised masking strategy is adopted: a subset of originally observed entries is randomly masked and treated as missing targets. The diffusion process is applied only to the missing part X_0^{mi}, while X_0^{ob} remains fixed and serves as conditioning context.

At diffusion step t, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ is injected into the missing component

$$X_t^{\text{mi}} = \sqrt{\bar{\alpha}_t}\, X_0^{\text{mi}} + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \tag{9}$$

where $\{\alpha_t\}_{t=1}^T$ is a predefined variance schedule and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. The model is trained to predict the injected noise using a conditional noise predictor $\epsilon_\theta(X_t^{\text{mi}}, t \mid X_0^{\text{ob}}, M)$. The denoising objective is minimized. Since diffusion is applied only to the missing component, the loss is computed exclusively on the missing positions

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{v,\ell}(1-M_{v,\ell})} \sum_{v=1}^{V}\sum_{\ell=1}^{L}(1-M_{v,\ell})\left(\epsilon_{v,\ell} - \epsilon_\theta(\cdot)_{v,\ell}\right)^2\right]. \tag{10}$$

The self-supervised training procedure of CTD-MTSI is summarized in Algorithm 1.

Algorithm 1 Training of CTD-MTSI

1: **Input:** training data distribution $q(X_0)$, number of iterations N_{iter}, noise schedule $\{\alpha_t\}_{t=1}^T$ with $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$
2: **Output:** trained conditional noise predictor ϵ_θ
3: **for** $i = 1$ **to** N_{iter} **do**
4: Sample $t \sim \text{Uniform}(\{1, \ldots, T\})$, $X_0 \sim q(X_0)$
5: Sample point-wise random masking on observed entries by updating M (e.g., set 30% of ones in M to zero)
6: Compute conditional context and missing targets:

$$X_0^{\text{ob}} \leftarrow M \odot X_0, \qquad X_0^{\text{mi}} \leftarrow (1-M) \odot X_0$$

7: Sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ with the same shape as X_0^{mi}
8: Apply forward noising to the missing part: $X_t^{\text{mi}} \leftarrow \sqrt{\bar{\alpha}_t}\, X_0^{\text{mi}} + \sqrt{1-\bar{\alpha}_t}\, \epsilon$
9: Predict noise using CTD-MTSI: $\hat{\epsilon} \leftarrow \epsilon_\theta\big(X_t^{\text{mi}},\, t \mid X_0^{\text{ob}},\, M\big)$
10: Take a gradient step minimizing the denoising objective (computed on missing entries):

$$\nabla_\theta \left(\frac{1}{\sum_{v,\ell}(1-M_{v,\ell})} \sum_{v=1}^{V}\sum_{\ell=1}^{L}(1-M_{v,\ell})\left(\epsilon_{v,\ell} - \hat{\epsilon}_{v,\ell}\right)^2\right)$$

11: **end for**

The model is optimized using Adam with an initial learning rate of 10^{-3} and a batch size of 16. Training runs for 200 epochs. A multi-step learning rate scheduler is employed, reducing the learning rate by 90% at 75% and 90% of the total training epochs.

Algorithm 2 Imputation (Conditional Reverse Sampling) with CTD-MTSI

Input: partially observed sample X_0, mask $M \in \{0,1\}^{V \times L}$, trained ϵ_θ
Output: imputed missing values $\hat{X}_0^{\text{mi}}$
Compute observed context: $X_0^{\text{ob}} \leftarrow M \odot X_0$
Initialize missing part with Gaussian noise: $X_T^{\text{mi}} \sim \mathcal{N}(0, I)$
for $t = T$ **down to** 1 **do**
Predict noise: $\hat{\epsilon} \leftarrow \epsilon_\theta\big(X_t^{\text{mi}},\, t \mid X_0^{\text{ob}},\, M\big)$
Compute DDPM mean: $\mu_\theta \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(X_t^{\text{mi}} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\hat{\epsilon}\right)$
Sample reverse step: $z \sim \mathcal{N}(0, I)$, $X_{t-1}^{\text{mi}} \leftarrow \mu_\theta + \sigma_t z$
end for
Set $\hat{X}_0^{\text{mi}} \leftarrow X_0^{\text{mi}}$ and return the full imputed series: $\hat{X}_0 \leftarrow X_0^{\text{ob}} + \hat{X}_0^{\text{mi}}$

A.3 Additional Qualitative Results

We provide additional qualitative imputation visualizations under varying point wise missing ratios (10%, 30%, 50%, and 70%) for both ETTh1 and ETTh2. These figures illustrate median predictions

and 95% confidence intervals, highlighting reconstruction accuracy and uncertainty calibration under increasing sparsity.

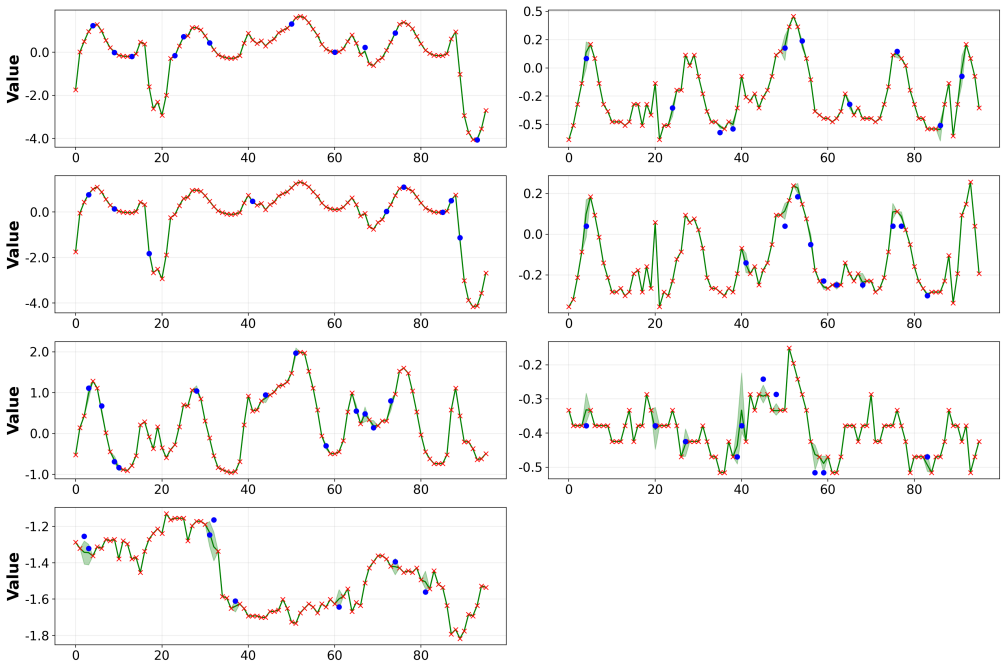

Figure 2: Visualization of probabilistic imputations results on ETTh1 (10% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

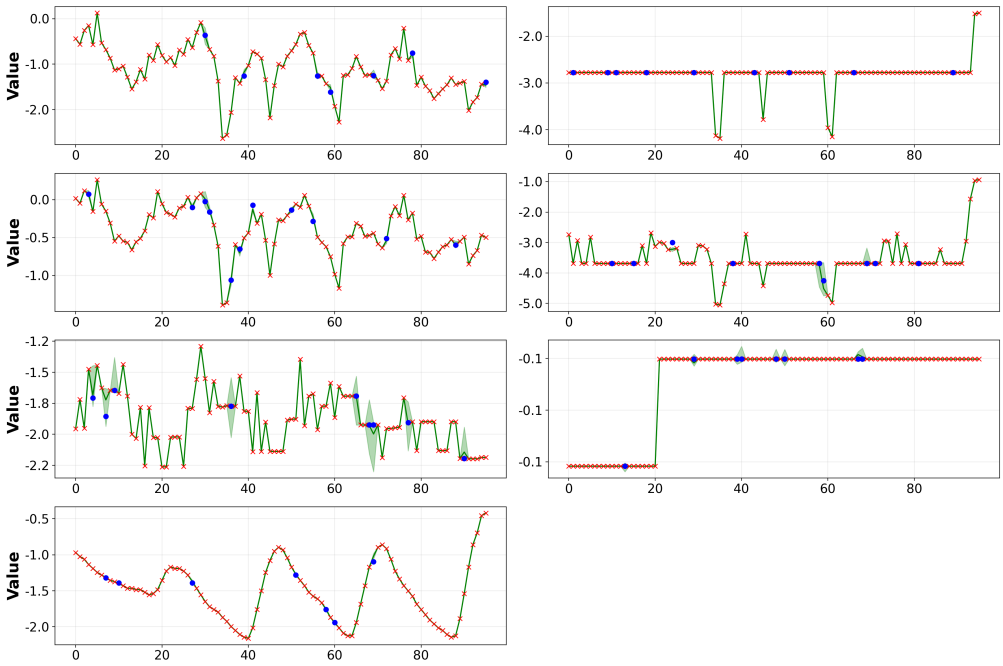

Figure 3: Visualization of probabilistic imputations results on ETTh2 (10% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

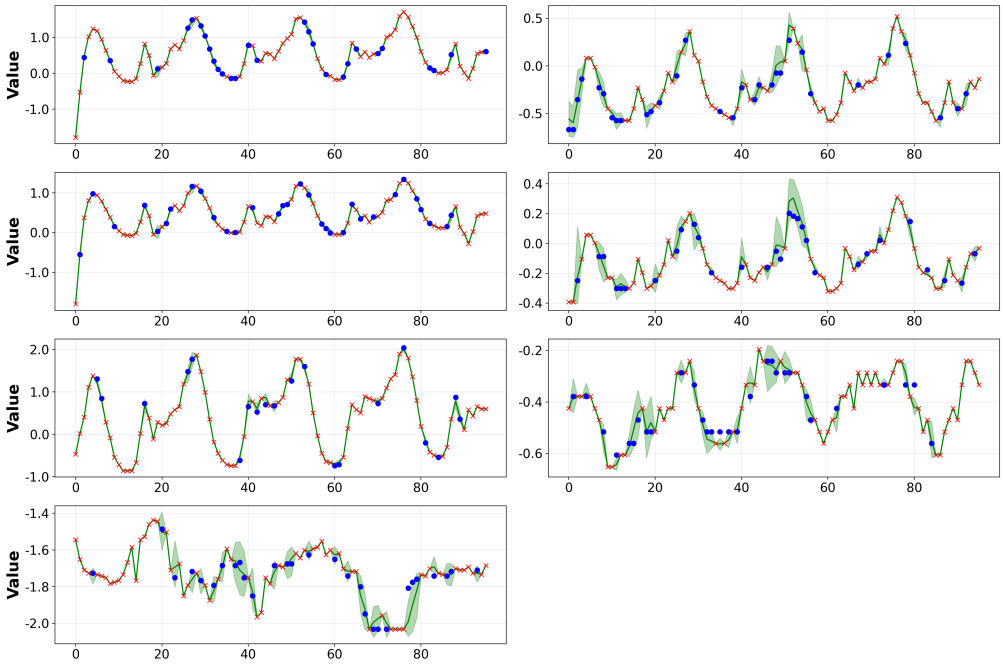

Figure 4: Visualization of probabilistic imputations results on ETTh1 (30% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

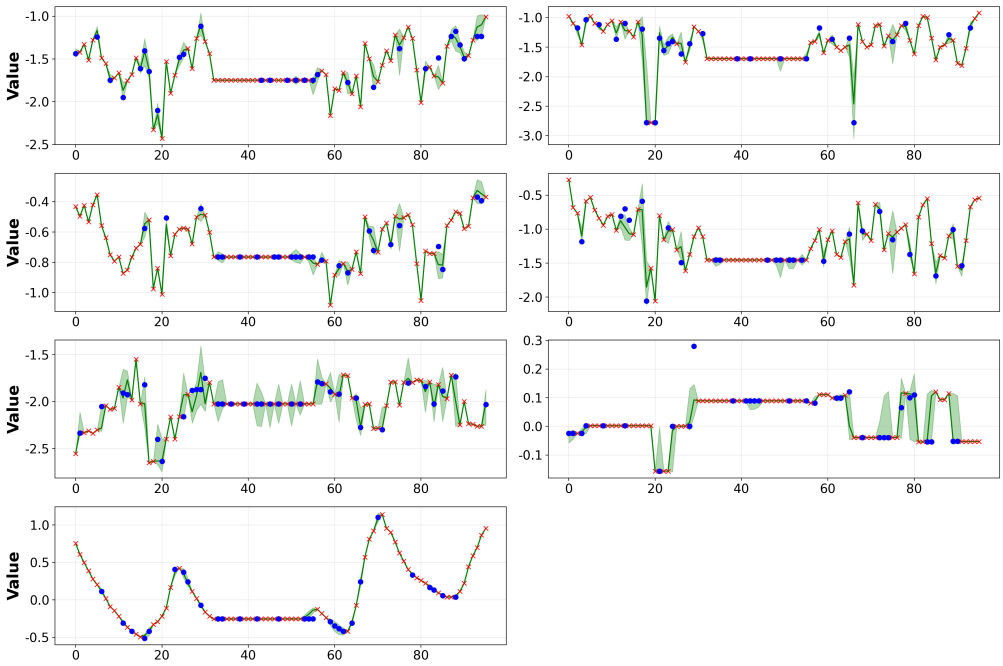

Figure 5: Visualization of probabilistic imputations results on ETTh2 (30% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

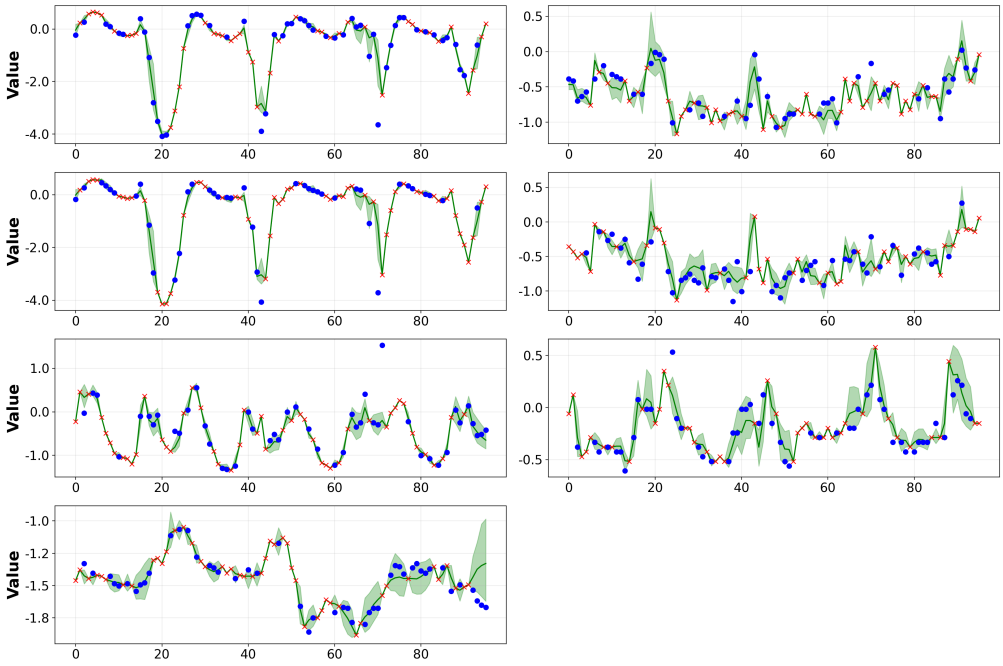

Figure 6: Visualization of probabilistic imputations results on ETTh1 (50% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

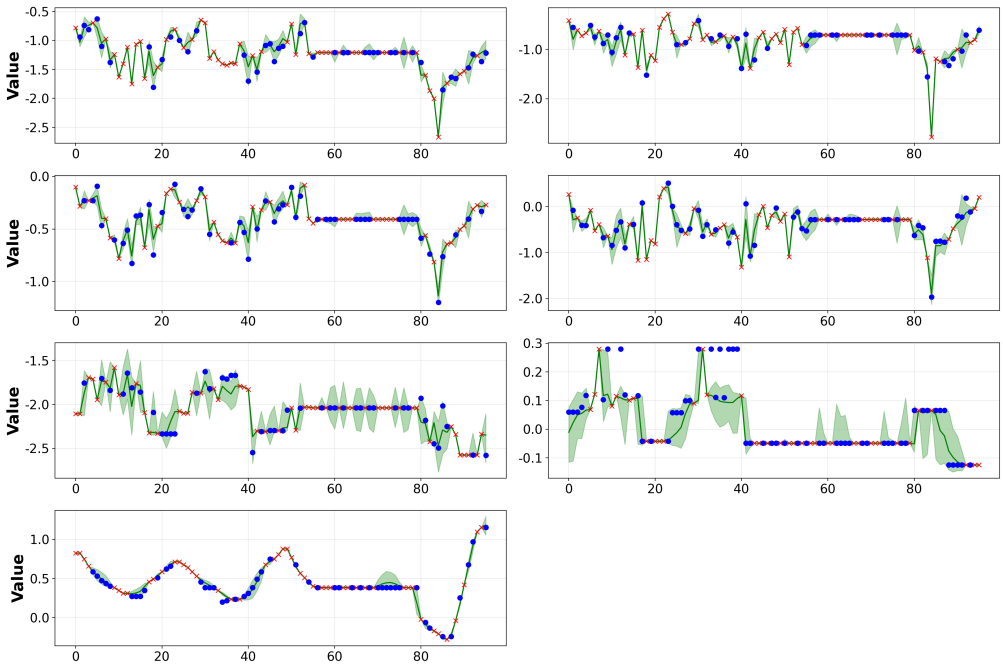

Figure 7: Visualization of probabilistic imputations results on ETTh2 (50% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

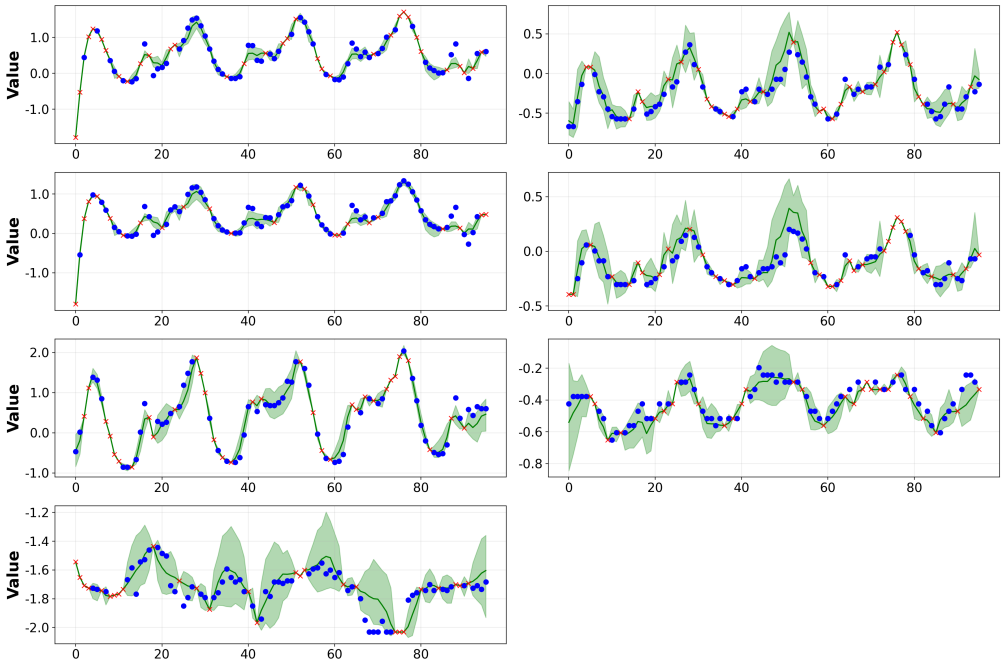

Figure 8: Visualization of probabilistic imputations results on ETTh1 (70% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

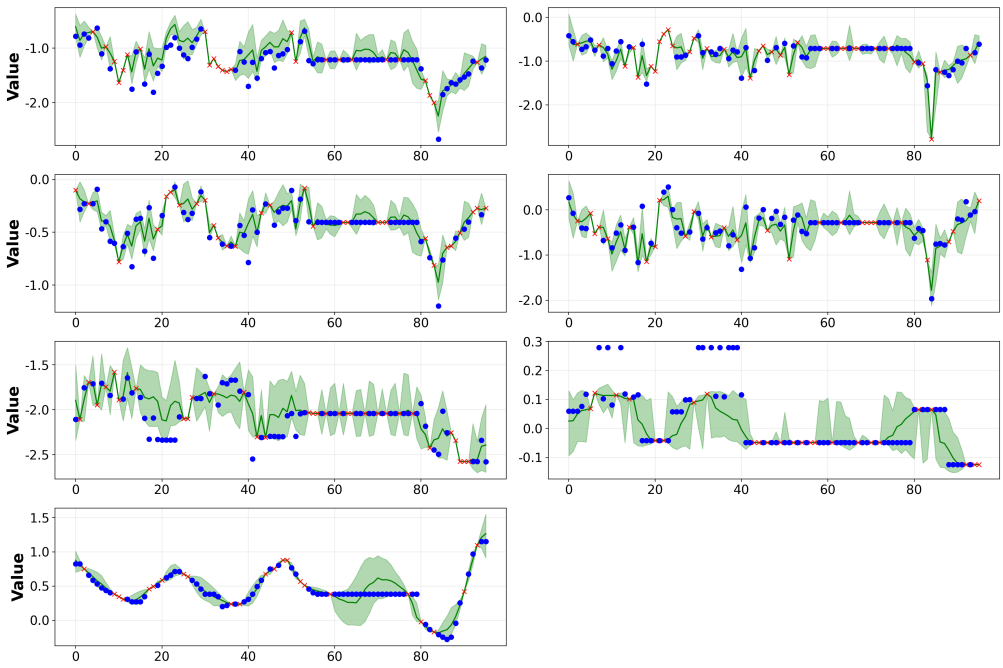

Figure 9: Visualization of probabilistic imputations results on ETTh2 (70% missingness). The results is for a time series sample with all 7 features. The median and 95% CIs are shown in green, observed points in red and targets in blue.

A.4 Noise Predictor Architecture Diagram

Figure 10 illustrates the overall architecture of the CTD-MTSI noise predictor. The model receives the noisy data X_t^{mi}, the observed data X_0^{ob}, and the diffusion timestep t as inputs. The backbone consists of (i) a mask-aware convolutional patch embedding stem, (ii) FiLM-based conditioning, (iii) hierarchical CNN–transformer modules combining depthwise temporal convolutions and variable attention, and (iv) a reconstruction head that outputs the predicted noise ϵ_θ.

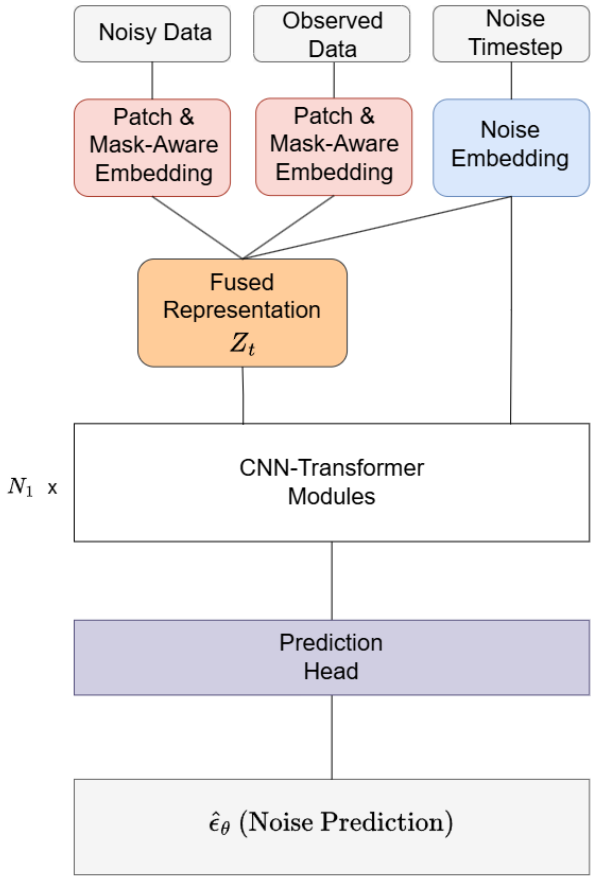

Figure 10: Diagram of the CTD-MTSI noise predictor backbone.

To provide further detail, Figure 11 illustrates the internal structure of a single CNN–transformer module used in the hierarchical backbone, combine with temporal downsampling. The module alternates depthwise temporal mixing and variable attention, followed by a convolutional feed-forward network, with noise-conditioned normalization applied at each residual branch.

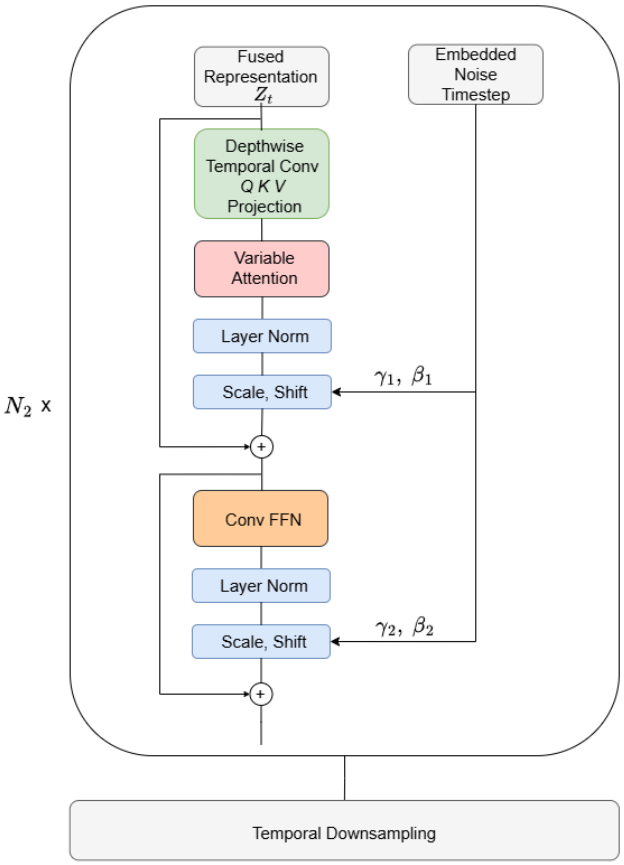

Figure 11: Diagram of the CNN-transformer module.