DiTS: Multimodal Diffusion Transformers Are Time Series Forecasters

Anonymous authors
Paper under double-blind review

Abstract

While generative modeling facilitates probabilistic time series forecasting, incorporating heterogeneous exogenous information remains challenging. Diffusion Transformers (DiT) provide a scalable framework for conditional generation, yet their adaptation to forecasting calls for conditioning mechanisms tailored to time series. Endogenous targets and exogenous covariates differ in sources, semantics, and statistical characteristics, while sharing temporal coordinates that support fine-grained conditional guidance. Covariates can describe future variability and temporal dependence beyond the conditional mean targeted by direct regression. Motivated by these considerations, we propose Diffusion Transformers for Time Series (DiTS), a Multimodal Diffusion Transformer for covariate-aware forecasting. DiTS models endogenous targets and exogenous covariates as distinct modalities, jointly conditioning future generation on target history and available covariates. Flow matching makes covariate-dependent distributional information relevant to velocity prediction conditioned on noisy future states. We introduce *Time-aligned Modulation*, extending AdaLN with the temporal-alignment prior to generate patch-wise modulation parameters from aligned covariates and diffusion time. Across diverse covariate-aware forecasting tasks, DiTS achieves strong performance in both deterministic and probabilistic forecasting, demonstrating the effectiveness of conditional generation for both point and distributional forecasting.

1 Introduction

Time series forecasting is a cornerstone of decision-making, from system monitoring to energy management. The evolution of an endogenous variate is shaped by both internal temporal dependencies and diverse exogenous factors, making external context a valuable source of guidance for anticipating its future outcomes. Nevertheless, targets and covariates often originate from different sources, with distinct semantics, units, and statistical characteristics. This modality-level gap calls for preserving their individual characteristics while modeling their dependencies. Meanwhile, regularly sampled multivariate series typically share a common clock, providing a natural temporal correspondence for integrating heterogeneous information.

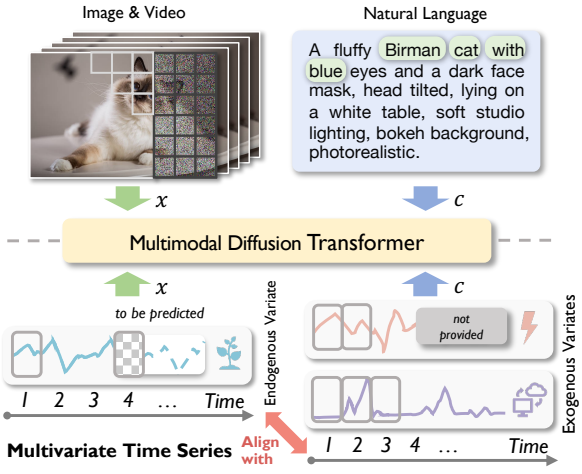


Figure 1: Multimodal generation and covariate-aware forecasting. DiTS treats endogenous targets and exogenous covariates as distinct modalities, exploiting their temporal alignment for fine-grained conditional control.

The development of deep forecasting models has been largely characterized by deterministic frameworks, with Transformers emerging as a prominent paradigm. Representative approaches refine temporal patch modeling (Nie et al., 2023), represent variates as tokens (Liu et al., 2024a), or distinguish endogenous modeling from exogenous aggregation (Wang et al., 2024). However, providing access to covariates does not

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.

by itself determine which aspects of their information the model learns to use. Changing external conditions can shift future distributions beyond what target history suggests. Covariates help account for these shifts, but direct squared-error regression targets only their effect on the conditional mean, leaving changes in variability and temporal dependence outside its prediction target.

Recently, Diffusion Transformers (Peebles & Xie, 2023) have provided a scalable backbone for conditional generation, with multimodal variants enabling fine-grained interaction between distinct input representations (Esser et al., 2024). This perspective extends to target generation guided by exogenous covariates (Figure 1). We adopt conditional flow matching (Lipman et al., 2022; Liu et al., 2022a) to learn a velocity field conditioned on target history and observed covariates, using probability paths from noise to future targets. At intermediate states, the optimal velocity depends on the expected clean future given the current state and these conditions. Covariate-dependent distributional differences can therefore affect velocity prediction even when the original conditional mean is unchanged, providing supervision beyond direct mean regression.

The common temporal coordinates of multivariate series offer a basis for adapting Multimodal Diffusion Transformers to forecasting. Corresponding target and covariate patches cover the same interval, providing a strong prior for where external information should enter. Rather than broadcasting a series-level representation across the window, AdaLN (Peebles & Xie, 2023) can exploit this correspondence to generate patch-specific parameters from aligned covariates and diffusion time. This motivates *Time-aligned Modulation*: temporal position determines where conditioning applies, while diffusion time specifies the generation stage. Temporal attention propagates conditioned information across positions, accommodating lagged dependencies without adding covariate tokens to attention.

To this end, we propose Diffusion Transformers for Time Series (**DiTS**), a Multimodal Diffusion Transformer integrating Time-aligned Modulation with conditional flow matching. DiTS treats endogenous targets and exogenous covariates as distinct modalities with different sources, semantics, and statistical characteristics but shared temporal coordinates. Generation is jointly conditioned on target history and available covariates: target patches retain historical context, while a *Covariate Encoder* preserves patch-wise external information. Time-aligned Modulation connects these representations, and DiTS blocks refine target patches through modulated temporal attention and feed-forward transformations. The learned velocity field supports probabilistic generation and sample-averaged point prediction. Experiments demonstrate strong forecasting performance, particularly with known future covariates, while ablations examine the contributions of conditional velocity learning and time-aligned injection. Our main contributions are summarized as follows:

- We propose DiTS, a Multimodal Diffusion Transformer for covariate-aware forecasting, using conditional flow matching to incorporate exogenous information into forecast generation and improve predictive accuracy over direct mean regression on the evaluated datasets.
- We introduce Time-aligned Modulation, extending AdaLN with patch-wise parameters generated from temporally aligned covariates. With a Covariate Encoder and temporal attention, it enables fine-grained conditional control without expanding the attention sequence.
- DiTS achieves strong performance across diverse covariate-aware forecasting tasks, attaining leading aggregate results in both deterministic and probabilistic forecasting. Ablation studies further validate its covariate utilization and generation efficiency.

2 Related Works

2.1 Forecasting with Exogenous Variables

Forecasting with exogenous variables requires relating target history to external observations that may be available only in the past or known over the prediction horizon. General multivariate Transformers organize temporal and cross-variate dependencies through channel-independent patches in PatchTST (Nie et al., 2023), variate tokens in iTransformer (Liu et al., 2024a), separate temporal and cross-dimension interactions in Crossformer (Zhang & Yan, 2022), or autoregressive multivariate patches in Timer-XL (Liu et al., 2024b). Methods designed for exogenous forecasting make the roles of targets and covariates more explicit. TFT (Lim et al., 2021) uses variable selection and attention to integrate heterogeneous inputs for multi-horizon prediction, while TimeXer (Wang et al., 2024) models endogenous series and aggregates exogenous information asymmetrically. DAG (Qiu

et al., 2026) discovers and injects temporal and channel correlations involving historical and future exogenous variables. GCGNet (Li et al., 2026) combines a variational generator, graph-structure alignment, and refinement to model correlations across time and variables in exogenous-aware forecasting. These advances highlight the importance of organizing information at an appropriate granularity. For covariate-aware forecasting, however, external information can describe more than changes in the future mean. Preserving its temporal correspondence and learning its distributional implications provide complementary opportunities beyond refining tokenization or attention alone.

2.2 Diffusion Models for Time Series

Early diffusion approaches include TimeGrad (Rasul et al., 2021a) for autoregressive forecasting and CSDI (Tashiro et al., 2021) for probabilistic imputation. Subsequent explorations incorporate structured state-space models in SSSD (Alcaraz & Strodthoff, 2022), latent diffusion in LDT (Feng et al., 2024), and general-purpose diffusion Transformers in TimeDiT (Cao et al., 2024), exploring representations and backbones for time series generation. Recently, Sundial (Liu et al., 2025) combines autoregressive context with flow-matching prediction. Beyond diffusion, TempFlow (Rasul et al., 2021b) uses conditioned normalizing flows for multivariate probabilistic forecasting, offering an alternative framework for capturing predictive distributions and cross-variate dependencies.

Beyond quantifying uncertainty, conditional generation makes distributional information relevant to forecasting. In flow matching, the same noisy state may require different updates under different exogenous conditions, even when their original conditional means agree. This motivates learning how covariates influence trajectories, complementing the development of expressive generative backbones.

2.3 Diffusion Transformers

Adopting Transformers has advanced scalability and unification in generative modeling. DiT (Peebles & Xie, 2023) introduces AdaLN-Zero to incorporate timestep and class information through adaptive normalization. PixArt-α (Chen et al., 2023) adopts cross-attention for text conditioning, while Multimodal Diffusion Transformers (Esser et al., 2024) combine rectified flow with modality-specific representations and joint attention. These developments demonstrate the value of accommodating heterogeneous inputs while maintaining fine-grained conditional control over generation.

Their adaptation to time series can exploit the temporal coordinates shared by targets and covariates. DiTS develops Time-aligned Modulation to preserve this correspondence within AdaLN, supplying each target patch with its covariate-dependent parameters. Temporal attention carries the conditioned information across patches, combining local conditional control with flexible temporal modeling.

3 Approach

We introduce DiTS, a Multimodal Diffusion Transformer for covariate-aware forecasting. As illustrated in Figure 2, target patches and a Covariate Encoder represent both modalities separately. Time-aligned Modulation injects aligned external information into temporal attention and feed-forward transformations, while conditional flow matching supports training and probabilistic forecasting.

3.1 Preliminaries

Let $\mathbf{x}_{0,\mathrm{his}} \in \mathbb{R}^W$ denote the historical observations of a target series and $\mathbf{y}_{0,\mathrm{pred}} \in \mathbb{R}^H$ its future values. The subscript 0 identifies the target, and $1{:}C$ indexes the exogenous covariates. We consider forecasting with both known future and past-only covariates, using only observations available at forecast time. For clarity, the following equations describe the known-future case, with $\mathbf{x}_{1:C} \in \mathbb{R}^{T\times C}$ and $T = W + H$. DiTS learns the conditional distribution of future targets given their history and the available exogenous observations.

We adopt the flow matching paradigm, specifically rectified flow (Lipman et al., 2022; Liu et al., 2022a), to learn a continuous transformation from Gaussian noise to the data distribution. Given independent noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_H)$, we define a linear interpolation between noise and future targets:

$$\mathbf{x}_{0,\mathrm{pred}}(t) = (1-t)\boldsymbol{\epsilon} + t\mathbf{y}_{0,\mathrm{pred}}, \qquad t \in [0,1]. \tag{1}$$

This formulation defines straight paths from noise at $t = 0$ to data at $t = 1$. We learn a network velocity field $\mathbf{v}_\theta$ conditioned on target history and exogenous covariates to approximate the corresponding time-dependent vector field.

The training objective minimizes the mean squared error between the predicted velocity and the target drift. Since the path is linear, the target velocity is $\mathbf{y}_{0,\text{pred}} - \boldsymbol{\epsilon}$:

$$\mathcal{L}_{\text{velocity}}(\theta) = \mathbb{E}\left[\left\|\mathbf{v}_\theta(\mathbf{x}_{0,\text{pred}}(t), t; \mathbf{x}_{0,\text{his}}, \mathbf{x}_{1:C}) - (\mathbf{y}_{0,\text{pred}} - \boldsymbol{\epsilon})\right\|_2^2\right]. \tag{2}$$

The expectation is over training windows, noise, and independently sampled diffusion times. We use logit-normal time sampling, $t = \text{sigmoid}(u)$ with $u \sim \mathcal{N}(0, 1)$. The path and loss use the normalized coordinates defined below; history and covariates remain observed conditions. Once trained, the model generates samples by solving the ordinary differential equation defined by the learned velocity field, integrating from Gaussian noise at $t = 0$ toward $t = 1$ without observing future target values.

3.2 Time Series Tokenization

Re-Normalization To account for scale variation across forecasting windows, we apply sample-wise normalization (Liu et al., 2022b). Target history and training targets use statistics computed exclusively from $\mathbf{x}_{0,\text{his}}$. Each covariate is normalized separately over its available observations: $W + H$ values for known future covariates and W for past-only covariates. We use population variances and add a stabilizer $\varepsilon = 10^{-5}$ before taking the square root. These statistics remain fixed during sampling. The following symbols denote normalized sequences; generated forecasts are subsequently restored to the target scale.

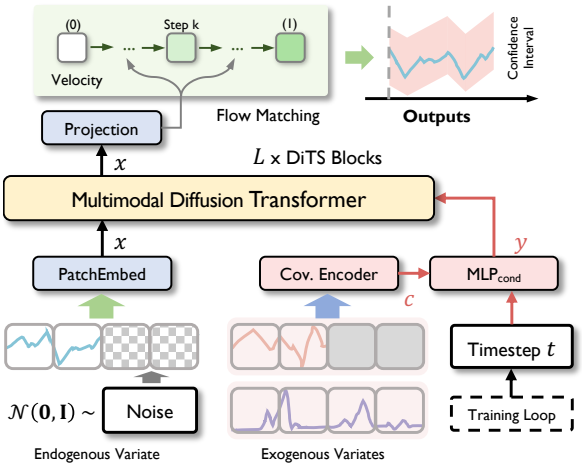


Figure 2: DiTS overview. Target patches pass through L DiTS blocks conditioned by aligned covariates and diffusion time via Time-aligned Modulation. The projected velocity guides forecast generation.

Patch Embedding We concatenate target history and the current future state as $\mathbf{x}_0(t) = [\mathbf{x}_{0,\text{his}}; \mathbf{x}_{0,\text{pred}}(t)] \in \mathbb{R}^T$. Historical observations remain fixed, while the future state follows Equation 1 during training and numerical integration during sampling. With W and H divisible by patch length P, we obtain $N = T/P$ non-overlapping patches, comprising $N_{\text{his}} = W/P$ historical and $N_{\text{pred}} = H/P$ future patches. A shared linear map produces their embeddings:

$$\mathbf{Z}^0_{x,n} = \text{PatchEmbed}\left(\mathbf{x}_0^{(n)}(t)\right) = \mathbf{W}_{\text{emb}}\mathbf{x}_0^{(n)}(t) + \mathbf{b}_{\text{emb}}, \qquad n = 1, \ldots, N, \tag{3}$$

where $\mathbf{x}_0^{(n)}(t) \in \mathbb{R}^P$, $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{D\times P}$, and $\mathbf{b}_{\text{emb}} \in \mathbb{R}^D$. Historical and future patches share the same embedding parameters, yielding $\mathbf{Z}^0_x \in \mathbb{R}^{N\times D}$ in temporal order.

Covariate Encoder Considering the heterogeneity of endogenous and exogenous variates, we encode covariates separately while preserving temporal correspondence with target patches. As shown in Figure 3, the C covariate patches at position n form $\mathbf{X}_c^{(n)} \in \mathbb{R}^{C\times P}$ with the same boundaries as the target. Flattening in a fixed variate order and applying two shared linear layers gives

$$\mathbf{c}_n = \text{Linear}_{c,2}\Big(\text{SiLU}\Big(\text{Linear}_{c,1}\Big(\text{Flatten}(\mathbf{X}_c^{(n)})\Big)\Big)\Big). \tag{4}$$

This yields $\mathbf{c} = [\mathbf{c}_1; \ldots; \mathbf{c}_N] \in \mathbb{R}^{N\times D}$, retaining a separate condition for each temporal patch. The encoder shares weights across positions, with fixed variate order preserving covariate identities. Its output is reused across DiTS blocks and integration steps; diffusion-time embedding and modulation-parameter generation are performed by Time-aligned Modulation. When future covariates are

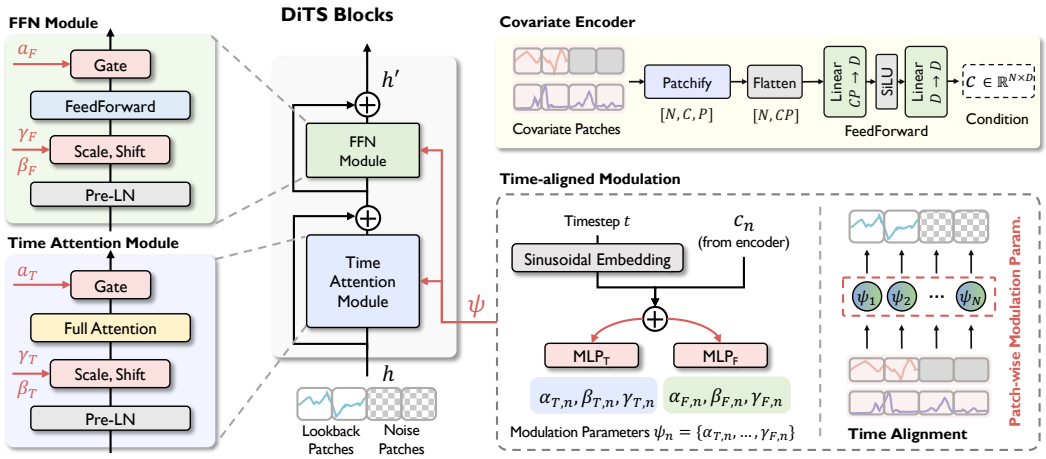


Figure 3: DiTS blocks and Time-aligned Modulation. Aligned covariate conditions $\mathbf{c}_n$ combine with diffusion time to generate patch-wise parameters ψ_n for scale, shift, and residual gates. This exploits temporal correspondence for local conditional control rather than broadcasting a single global condition across the window, while Time Attention propagates information across target patches.

unavailable, we first encode the historical covariate patches and use a shared linear adapter along the temporal axis to map their representations to the future patch positions. Concatenating these projected conditions with the historical conditions yields the same aligned sequence of N covariate representations; all subsequent DiTS operations are identical to the known-future case.

3.3 DiTS Block

The DiTS block combines temporal modeling and covariate injection through Time-aligned Modulation (Figure 3). It contains temporal attention and a point-wise FFN, both guided by aligned covariate conditions. Its input h and output h' denote the target representations $\mathbf{Z}_x^{l-1}$ and $\mathbf{Z}_x^l$.

Time-aligned Modulation Target and covariate patches at position n cover the same interval. Exploiting this prior, each block generates patch-wise modulation parameters from $\mathbf{c}_n$ and the diffusion-time embedding $\mathbf{e}_{\text{time}}(t) \in \mathbb{R}^D$, obtained by applying an MLP to a sinusoidal encoding of t:

$$\left[\boldsymbol{\alpha}^l_{m,n}(t), \boldsymbol{\beta}^l_{m,n}(t), \boldsymbol{\gamma}^l_{m,n}(t)\right] = \text{MLP}^l_m(\mathbf{c}_n + \mathbf{e}_{\text{time}}(t)), \quad m \in \{T, F\}. \tag{5}$$

The D-dimensional vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ are the residual gate, shift, and scale offset for temporal attention (T) or the FFN (F). We collect them as

$$\psi^l_n(t) = \left\{\boldsymbol{\alpha}^l_{T,n}(t), \boldsymbol{\beta}^l_{T,n}(t), \boldsymbol{\gamma}^l_{T,n}(t), \boldsymbol{\alpha}^l_{F,n}(t), \boldsymbol{\beta}^l_{F,n}(t), \boldsymbol{\gamma}^l_{F,n}(t)\right\}. \tag{6}$$

We write ψ_n when l, t are implicit and collect the N sets as $\Psi^l(t) = (\psi^l_1(t), \ldots, \psi^l_N(t))$. *Patch-wise Modulation Parameters* denotes separately generated values at each position, using weights shared across positions within each block. The two heads are implemented together by SiLU and a zero-initialized $D \to 6D$ projection. Consequently, modulation responds to both the local covariate context and the current generation stage, preserving the distinct roles of sequence time and diffusion time.

For $\mathbf{H} \in \mathbb{R}^{N \times D}$, modulation acts on each aligned patch:

$$[\text{Mod}^l_m(\mathbf{H})]_n = (\mathbf{1} + \boldsymbol{\gamma}^l_{m,n}(t)) \odot \text{LN}^l_m(\mathbf{H}_n) + \boldsymbol{\beta}^l_{m,n}(t), \tag{7}$$

with non-affine LayerNorm following AdaLN-Zero (Peebles & Xie, 2023). The scale and shift adapt the normalized features according to each patch's temporally aligned covariate context, while the residual gate controls the module's contribution. Zero-initialized gates make each block an identity map at initialization. Below, t is implicit and $\boldsymbol{\alpha}^l_m \in \mathbb{R}^{N \times D}$ stacks the patch-wise gates.

Time Attention and FFN Each block applies temporal attention followed by a point-wise FFN, with Time-aligned Modulation conditioning both transformations. Patch-wise residual gates control their contributions to the target representations:

$$\begin{aligned}\hat{\mathbf{Z}}_x^l &= \mathbf{Z}_x^{l-1} + \boldsymbol{\alpha}_T^l \odot \mathrm{MSA}_{\mathrm{time}}^l\Big(\mathrm{Mod}_T^l(\mathbf{Z}_x^{l-1})\Big), \\ \mathbf{Z}_x^l &= \hat{\mathbf{Z}}_x^l + \boldsymbol{\alpha}_F^l \odot \mathrm{FFN}^l\Big(\mathrm{Mod}_F^l(\hat{\mathbf{Z}}_x^l)\Big).\end{aligned} \tag{8}$$

Temporal attention uses Q/K RMS normalization and rotary positional embeddings, with full connectivity over historical and noisy future patches. All future coordinates are updated jointly, without causal masking. Since modulation precedes attention, covariates can influence temporal aggregation beyond their aligned positions, accommodating lagged dependencies. The FFN applies $D \to D_{\mathrm{ff}} \to D$ linear maps with GELU to each token. Only target representations evolve through the blocks; covariates supply modulation conditions. This keeps attention over N target tokens, with pairwise cost $\mathcal{O}(N^2 D)$, while the Covariate Encoder's input projection grows linearly with the number of covariates. Fine-grained conditions therefore do not require attention over an expanded time–variate token grid.

3.4 DiTS Backbone

The backbone stacks L DiTS blocks, each conditioned by the modulation parameter sets $\Psi^l(t)$:

$$\mathbf{Z}_x^l = \text{DiTS-Block}_l\big(\mathbf{Z}_x^{l-1}, \Psi^l(t)\big), \qquad l = 1, \ldots, L. \tag{9}$$

The Covariate Encoder output $\mathbf{c}$ is shared across blocks, while each block generates its own modulation parameters. Successive layers refine target representations through the interaction of aligned external conditions and temporal context. A zero-initialized linear head then projects the final future tokens to patch-wise velocities:

$$\mathbf{v}_\theta = \mathrm{Unpatchify}\big(\mathrm{Linear}_{\mathrm{out}}\big(\mathbf{Z}_{x,N_{\mathrm{his}}+1:N}^L\big)\big) \in \mathbb{R}^H. \tag{10}$$

Historical tokens provide context, and the head predicts velocities only for the future window. Training follows Equation 2, allowing covariates to guide state-dependent updates throughout generation.

At inference, K forward Euler steps integrate the learned velocity from noise to data. With $t_k = k/K$ and $\widehat{\mathbf{x}}_{0,\mathrm{pred}}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_H)$, the numerical state is updated by

$$\widehat{\mathbf{x}}_{0,\mathrm{pred}}^{(k+1)} = \widehat{\mathbf{x}}_{0,\mathrm{pred}}^{(k)} + \frac{1}{K}\mathbf{v}_\theta\Big(\widehat{\mathbf{x}}_{0,\mathrm{pred}}^{(k)}, t_k; \mathbf{x}_{0,\mathrm{his}}, \mathbf{x}_{1:C}\Big), \quad k = 0, \ldots, K-1. \tag{11}$$

Here, (k) indexes integration steps. Each step reuses the fixed history and covariate encoding, while recomputing future patch embeddings and time-dependent modulation parameters. Historical hidden representations are recomputed within each backbone evaluation and are not part of the integration state. Starting from independent noise draws, we generate S trajectories, restore their original scale, and use their average for point forecasting. The samples also characterize predictive uncertainty.

4 Experiments

We extensively evaluate DiTS on covariate-aware forecasting, considering both point accuracy and probabilistic forecast quality. Experiments cover five electricity price forecasting (EPF) datasets and 13 selected FEV-Bench tasks, followed by analyses of covariate utilization, condition control, and generation efficiency. We report MSE, MAE, and MASE for point forecasts, and WQL and CRPS for probabilistic evaluation; lower values are better. Dataset descriptions, implementation details, and full results are provided in Appendices A, B, and D.

4.1 Time Series Covariate-aware Forecasting

We evaluate DiTS on complementary deterministic and probabilistic forecasting tasks, examining both point accuracy and predictive distribution quality. Together, these evaluations assess covariate-conditioned generation across different information-availability settings, application domains, and temporal resolutions, with particular emphasis on the effective utilization of exogenous information.

Table 1: Deterministic EPF forecasting with known future covariates, averaged over $H = 24$ and $H = 360$. Lower is better; **bold** and underlined mark the best and second-best results. GCGNet results are from Li et al. (2026), and DAG results are from Qiu et al. (2026).

Model	DiTS		GCGNet		TimeXer		DAG		iTrans.		Cross.		DUET		TFT	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.290**	**0.315**	0.346	0.337	0.375	0.355	0.362	0.344	0.354	0.373	0.300	0.323	0.411	0.408	0.379	0.375
PJM	**0.085**	**0.171**	0.093	0.186	0.124	0.225	0.093	0.180	0.098	0.197	0.119	0.212	0.102	0.197	0.114	0.207
BE	**0.375**	**0.261**	0.423	0.287	0.432	0.279	0.423	0.279	0.513	0.345	0.403	0.268	0.515	0.354	0.454	0.291
FR	**0.332**	**0.202**	0.405	0.230	0.415	0.230	0.414	0.219	0.484	0.261	0.412	0.232	0.496	0.327	0.504	0.257
DE	**0.270**	**0.317**	0.387	0.387	0.406	0.391	0.370	0.370	0.408	0.407	0.356	0.376	0.482	0.430	0.489	0.446
AVG	**0.270**	**0.253**	0.331	0.285	0.350	0.296	0.332	0.278	0.371	0.317	0.318	0.283	0.401	0.343	0.388	0.315

Table 2: Deterministic EPF forecasting without future covariates at $(W = 168, H = 24)$; **bold** and underlined mark the best and second-best results. GCGNet and DAG results are from Li et al. (2026) and Qiu et al. (2026), respectively; the remaining baselines are from Wang et al. (2024).

Model	DiTS		GCGNet		TimeXer		DAG		iTrans.		PatchTST		Cross.		DLinear	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.218**	**0.257**	0.234	0.259	0.236	0.268	0.237	0.259	0.265	0.300	0.267	0.284	0.240	0.285	0.309	0.321
PJM	**0.069**	**0.161**	0.073	0.168	0.093	0.192	0.072	0.162	0.097	0.197	0.106	0.209	0.101	0.199	0.108	0.215
BE	0.370	0.263	**0.363**	0.251	0.379	**0.243**	0.376	0.244	0.394	0.270	0.400	0.262	0.420	0.290	0.463	0.313
FR	0.341	0.198	**0.338**	0.194	0.385	0.208	0.345	**0.190**	0.439	0.233	0.411	0.220	0.434	0.208	0.429	0.260
DE	**0.403**	**0.397**	0.415	0.400	0.440	0.415	0.422	0.411	0.479	0.443	0.461	0.432	0.574	0.430	0.520	0.463
AVG	**0.280**	0.255	0.285	0.254	0.307	0.265	0.290	**0.253**	0.335	0.289	0.330	0.282	0.354	0.282	0.366	0.314

Deterministic Forecasting We evaluate electricity price forecasting on five datasets—NP, PJM, BE, FR, and DE—using external forecasts as covariates (Lago et al., 2021). Generated trajectories are averaged for point prediction. With known future covariates, DiTS achieves the lowest MSE and MAE on each dataset when averaged over $(W, H) = (168, 24)$ and $(720, 360)$ (Table 1). Average MSE and MAE improve by more than 10% and 9.0%, respectively, over the strongest baseline for each metric, supporting conditional generation with temporally aligned exogenous guidance.

With past-only covariates at $(W, H) = (168, 24)$, DiTS retains the lowest average MSE and leads both metrics on NP, PJM, and DE (Table 2). Its average MAE remains competitive, although other methods lead on BE and FR. These results demonstrate effective point forecasting even when external information is restricted to history.

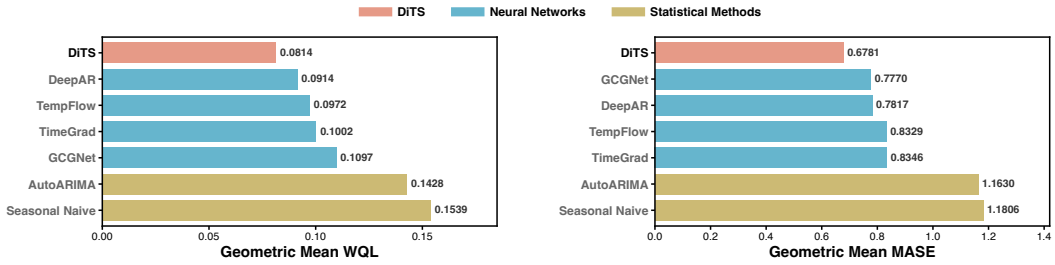
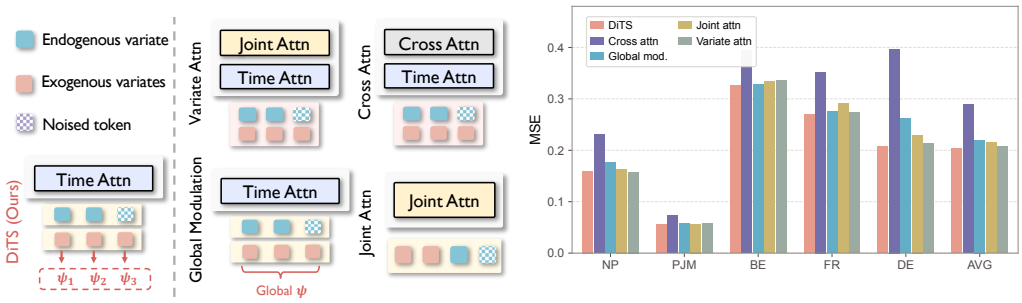


Figure 4: Probabilistic forecasting on 13 covariate datasets across domains and temporal resolutions. DiTS achieves the lowest geometric mean WQL and MASE in the reported comparison, supporting covariate-conditioned generation for both distributional modeling and point forecasting.

Probabilistic Forecasting We evaluate 13 tasks from Shchur et al. (2025), spanning electricity demand, prices, and solar generation. Baselines include DeepAR (Salinas et al., 2020), TimeGrad (Rasul et al., 2021a), TempFlow (Rasul et al., 2021b), and GCGNet (Li et al., 2026), alongside AutoARIMA (Hyndman & Khandakar, 2008) and Seasonal Naive (Hyndman & Athanasopoulos, 2018). All neural models receive known future covariates; DiTS uses sample medians and quantiles for point and probabilistic evaluation.

DiTS achieves the lowest geometric mean WQL and MASE, improving on the strongest competing aggregates by 10.9% and 12.7%, respectively (Figure 4). It performs particularly well on GFC tasks,



(a) Covariate-conditioning architectures.

(b) MSE across the EPF datasets.

Figure 6: Covariate-conditioning mechanisms and forecasting accuracy under $(W, H) = (168, 24)$. DiTS uses Time-aligned Modulation; the alternatives use cross attention, global modulation, joint attention, or variate attention. In every variant, diffusion time enters through AdaLN; this pathway is omitted from the diagrams for clarity. The right panel reports MSE for each EPF dataset and the five-dataset average. These results favor temporally aligned modulation over global conditioning.

while other models remain competitive on load and solar tasks (Table 11). These results support modeling covariate-dependent distributions while retaining accurate point forecasts, consistent with richer covariate supervision through noisy-state conditioning.

4.2 Model Analysis

Conditional Flow Matching We compare flow matching with direct MSE regression on EPF at $(W, H) = (168, 24)$, using the same backbone, architecture hyperparameters, and optimization protocol. Flow matching improves average MSE by 10.5% with known future covariates and 7.3% with past-only covariates, with gains across all five datasets (Figure 5, Table 10). Without covariates, it performs slightly worse than direct regression in this comparison. Relative to each method's target-only baseline, flow matching therefore benefits more from both types of covariates; past-only covariates provide no average improvement under direct regression at the reported precision.

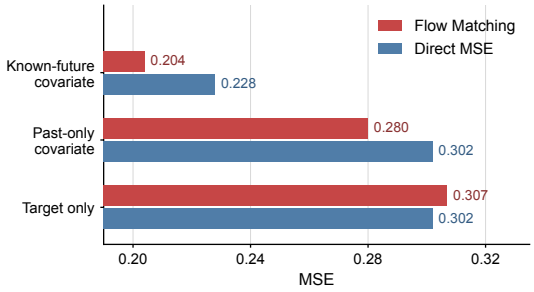


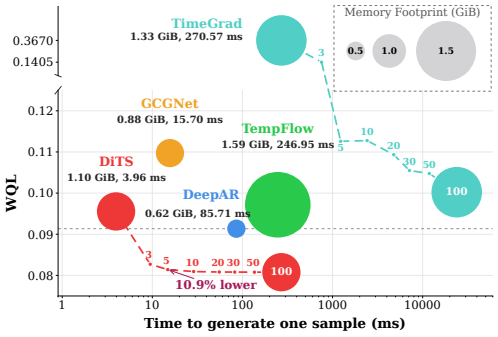
Figure 5: Flow matching versus direct MSE regression under three covariate settings. Bars report mean EPF MSE. Gains emerge with covariates, consistent with noisy-state conditioning making covariate-dependent distributional information relevant to velocity prediction.

These incremental gains are consistent with our analysis: given target history, covariates may convey distributional information beyond the conditional mean. Noisy-state conditioning can turn these differences into posterior-mean differences relevant to velocity prediction. Ignoring this information can increase flow-matching loss even when direct regression is unaffected, providing supervision that may improve covariate utilization.

Condition Control We compare Time-aligned Modulation with cross attention, global modulation, joint attention, and variate attention (Figure 6). Under $(W, H) = (168, 24)$, it achieves the lowest average MSE/MAE of 0.204/0.207 (Table 13). Relative to global modulation, average MSE decreases from 0.220 to 0.204, a 7.3% improvement, with gains in both metrics across all five datasets. Joint, variate, and cross attention obtain MSEs of 0.215, 0.207, and 0.289, respectively.

These results support exploiting temporal correspondence in covariate injection. Time-aligned Modulation supplies external guidance at matching target positions, while temporal attention propagates its influence across the window to accommodate cross-time dependencies. Although individual datasets favor different alternatives, DiTS obtains the lowest five-dataset average MSE and MAE among these conditioning mechanisms without a separate evolving covariate stream.

Model Efficiency Figure 7 compares WQL, generation latency, and training memory across 13 forecasting tasks, with labels indicating steps per trajectory. DiTS surpasses the baselines in WQL with three steps; further steps yield diminishing gains at higher latency. Its training memory lies below TimeGrad and TempFlow but above DeepAR and GCGNet. DiTS combines accurate forecasting with efficient few-step generation, balancing predictive performance and computational cost.



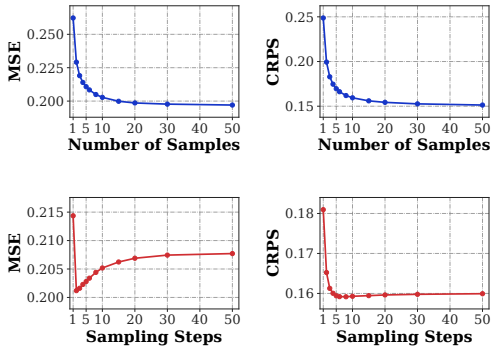


Figure 7: WQL versus time on 13 datasets. Bubble area shows training memory; dashed curves vary the number of sampling steps per trajectory.

Figure 8: Sample-count (top) and sampling step (bottom) sensitivity on five EPF datasets, fixing five steps and ten samples, respectively.

4.3 Sampling and Noise Scheduling

Inference Budget Figure 8 varies sample count S with five Euler steps and integration steps K with ten samples, using fixed checkpoints. More samples improve MSE and CRPS with diminishing returns. More steps initially improve CRPS before saturation, whereas MSE rises after its minimum. Higher-order solvers provide no improvement at five steps (Appendix C.3), favoring few steps and multiple trajectories. These comparisons distinguish two uses of computation: sampling stabilizes predictive statistics, while finer integration follows the learned velocity field more closely without directly optimizing point-forecast error.

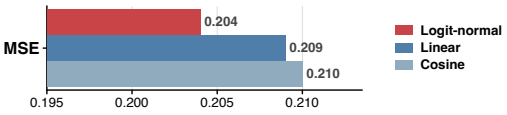


Figure 9: Training-time sampling on five EPF datasets. Logit-normal gives the lowest mean MSE among the selected configurations.

Training-Time Noise Schedule Training-time sampling determines how supervision is distributed across generation stages. We compare logit-normal, linear, and cosine schedules at $(W, H) = (168, 24)$, obtaining mean MSE of 0.204, 0.209, and 0.210 (Figure 9). Logit-normal leads on four datasets, while cosine leads on BE. The results favor logit-normal within this search, although the selected hyperparameters can contribute to the differences. Further comparisons of prediction parameterizations, noise paths, and guidance appear in Appendices C.1–C.4.

5 Conclusion

We introduced DiTS, a Multimodal Diffusion Transformer for time series generation conditioned on target history and covariates. Conditional flow matching makes covariate-dependent distributional information relevant to velocity prediction, providing supervision beyond direct mean regression. Time-aligned Modulation extends AdaLN with patch-wise parameters from aligned covariates and diffusion time, preserving temporal correspondence while temporal attention captures cross-time dependencies. Experiments demonstrate strong point and probabilistic forecasting, with ablations supporting covariate utilization under flow matching and the effectiveness of aligned modulation. DiTS achieves favorable quality with few sampling steps, balancing accuracy and generation cost. These findings support time-aligned conditional generation for covariate-aware forecasting.

AI USE STATEMENT

Generative AI tools were used solely to assist with translation, drafting, and revising portions of the manuscript, language polishing, literature search, reference formatting, and figure preparation. They were not used to develop theoretical models or conceptual frameworks, design research methodology or experiments, assist in writing proofs, implement experimental methods, generate or process datasets, support data analysis, or interpret results, nor were substantive AI-generated suggestions for mathematical assumptions, derivations, proofs, or research hypotheses adopted. All AI-assisted work was reviewed by the authors, who consulted original references, checked mathematical derivations and experimental code, and verified reported results against experimental records, and we take full responsibility for the final content of this work.

REFERENCES

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022.

Defu Cao, Wen Ye, Yizhou Zhang, and Yan Liu. TimeDiT: General-purpose diffusion transformers for time series foundation model. *arXiv preprint arXiv:2409.02322*, 2024.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-$alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, et al. Goku: Flow based video generative foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2502.04896.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, 27:1–22, 2008.

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. URL https://arxiv.org/abs/2511.13720.

Zhengyu Li, Xiangfei Qiu, Yuhan Zhu, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. GCGNet: Graph-consistent generative network for time series forecasting with exogenous variables. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2f27964513a28d034530bfdd117ea31d-Abstract-Conference.html.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4): 1748–1764, 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022a.

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

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2024a.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024b.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2023.

Adam Paszke, S. Gross, Francisco Massa, A. Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Z. Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Xiangfei Qiu, Yuhan Zhu, Zhengyu Li, Xingjian Wu, Bin Yang, and Jilin Hu. DAG: A dual correlation network for time series forecasting with exogenous variables. *arXiv preprint arXiv:2509.14933v3*, 2026.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pp. 8857–8868. PMLR, 2021a.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. 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`.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.*, 2020.

Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Yuyang Wang. fev-bench: A realistic benchmark for time series forecasting. *arXiv preprint arXiv:2509.26468*, 2025.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

A DATASETS

This section describes the Electricity Price Forecasting (EPF) datasets and 13 probabilistic forecasting tasks selected from Shchur et al. (2025), summarized in Tables 3 and 4, respectively.

A.1 PROBABILISTIC FORECASTING TASK SELECTION

We select tasks suitable for evaluating probabilistic forecasts with known-future covariates according to two criteria:

- **Known-future covariates:** Each task provides dynamic exogenous variates available over the forecast horizon.
- **Sufficient historical coverage:** Each task has enough history for end-to-end training and the rolling evaluation protocol under the computational budget of this study.

The five EPF datasets are also among these 13 tasks. Their rolling-window probabilistic evaluation differs from the setting used for deterministic forecasting on the EPF datasets (Wang et al., 2024).

A.2 ELECTRICITY PRICE FORECASTING DATASETS

The EPF collection (Lago et al., 2021) comprises five hourly day-ahead electricity-price datasets, each spanning approximately six years. Their exogenous inputs are forecasts available before the prediction horizon, such as load or generation forecasts, rather than realized future observations.

NP (Nord Pool) spans 2013-01-01 to 2018-12-24 and includes electricity prices, grid load forecasts, and wind power forecasts. PJM (Pennsylvania-New Jersey-Maryland) covers the same period, with prices in the Commonwealth Edison (COMED) zone and forecasts of system and COMED load. BE (Belgium) spans 2011-01-09 to 2016-12-31 and includes electricity prices, Belgian load forecasts, and generation forecasts from neighboring France. FR (France) spans 2012-01-09 to 2017-12-31 and includes electricity prices alongside load and generation forecasts. DE (Germany) spans 2012-01-09 to 2017-12-31 and includes electricity prices, load forecasts for the TSO Amprion zone, and wind and solar generation forecasts.

Table 3: Summary of the five Electricity Price Forecasting (EPF) datasets. The last column reports the numbers of sliding forecasting windows in the training, validation, and test splits, respectively, for $(W, H) = (168, 24)$.

Dataset	Freq.	Target Variate	Forecast Windows (Train, Val, Test)
NP	Hourly	Electricity Price	(36500, 5219, 10460)
PJM	Hourly	Electricity Price	(36500, 5219, 10460)
BE	Hourly	Electricity Price	(36500, 5219, 10460)
FR	Hourly	Electricity Price	(36500, 5219, 10460)
DE	Hourly	Electricity Price	(36500, 5219, 10460)

B IMPLEMENTATION DETAILS

Our model is implemented using PyTorch (Paszke et al., 2019), and all experiments are conducted on a single NVIDIA RTX 4090 GPU (24GB).

B.1 FLOW MATCHING AND INFERENCE CONFIGURATION

We sample training times from a logit-normal distribution, $t = \mathrm{sigmoid}(u)$ with $u \sim \mathcal{N}(0, 1)$, and optimize the velocity loss in Equation 2. Unless specified otherwise, inference uses five Euler integration steps per trajectory and averages ten independently initialized forecast samples for point prediction. The training-time schedule comparison is reported in Figure 9; the sample-count and integration-step sensitivities are shown in Figure 8. Evaluation on the 13 selected probabilistic tasks uses 50 forecast samples, as specified below.

Table 4: Summary of 13 probabilistic forecasting tasks selected from Shchur et al. (2025). H is the prediction horizon and N_{roll} is the number of rolling evaluation windows. Covariates are counted separately according to whether they are available only in the past or known over the forecast horizon.

Dataset	Freq.	H	N_{roll}	Median Length	# Ser.	# Past Cov.	# Known Cov.
ENTSO-e Load	15min	96	20	175,292	6	0	3
ENTSO-e Load	30min	96	20	87,645	6	0	3
ENTSO-e Load	Hourly	168	20	43,822	6	0	3
EPF BE	Hourly	24	20	52,416	1	0	2
EPF DE	Hourly	24	20	52,416	1	0	2
EPF FR	Hourly	24	20	52,416	1	0	2
EPF NP	Hourly	24	20	52,416	1	0	2
EPF PJM	Hourly	24	20	52,416	1	0	2
GFC12	Hourly	168	10	39,414	11	0	1
GFC14	Hourly	168	20	17,520	1	0	1
GFC17	Hourly	168	20	17,544	8	0	1
Solar with Weather	15min	96	20	198,600	1	2	7
Solar with Weather	Hourly	24	20	49,648	1	2	7

B.2 Hyperparameter Settings

We use separate hyperparameter grids for EPF and the 13 selected probabilistic tasks.

EPF Dataset. For EPF, we search the 12 combinations of learning rate in $\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$, model dimension $d_{\text{model}} \in \{256, 512\}$, and encoder depth $e_{\text{layers}} \in \{3, 4\}$. Batch size is 32, patch length is 24, and the feed-forward dimension is $d_{\text{ff}} = 4d_{\text{model}}$.

Selected Probabilistic Tasks. For each selected task, we evaluate DiTS under four $(d_{\text{model}}, d_{\text{ff}}, e_{\text{layers}})$ configurations: $(256, 768, 3)$, $(256, 1024, 4)$, $(512, 1536, 3)$, and $(512, 2048, 4)$, each with learning rates in $\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$. The patch length is 24 and the batch size is 64. Each baseline is evaluated over 12 hyperparameter configurations based on the settings reported in its original paper. For each task, all baselines use the same lookback length as DiTS. Table 5 reports the selected DiTS setting for all 13 tasks.

Table 5: Selected DiTS hyperparameters for the 13 probabilistic forecasting tasks. W is the history length; the patch length is 24 and the batch size is 64 throughout.

Dataset	Freq.	W	Batch	d_{model}	d_{ff}	LR	e_{layers}
ENTSO-e Load	15min	720	64	512	2048	10^{-3}	4
ENTSO-e Load	30min	720	64	512	2048	10^{-4}	4
ENTSO-e Load	Hourly	720	64	512	2048	10^{-3}	4
GFC12	Hourly	720	64	256	768	3×10^{-4}	3
GFC14	Hourly	720	64	256	1024	3×10^{-4}	4
GFC17	Hourly	720	64	256	768	10^{-3}	3
Solar with Weather	15min	720	64	512	1536	10^{-4}	3
Solar with Weather	Hourly	336	64	256	768	10^{-3}	3
EPF BE	Hourly	720	64	512	1536	10^{-3}	3
EPF DE	Hourly	720	64	512	2048	10^{-4}	4
EPF FR	Hourly	720	64	256	1024	3×10^{-4}	4
EPF NP	Hourly	720	64	512	2048	10^{-3}	4
EPF PJM	Hourly	720	64	256	768	10^{-4}	3

B.3 Protocols for Additional Experiments

We report additional ablations on prediction parameterization, generative noise paths, numerical solvers, classifier-free guidance, and hyperparameter sensitivity in Appendices C.1–C.5. Ablations requiring retraining use the 12-configuration EPF grid described above, and the reported results are

selected from this grid separately for each dataset and experimental setting. Table 10 separately compares covariate-availability settings.

C Additional Analysis

C.1 Prediction and Loss Parameterizations

In a flow-matching formulation, Li & He (2025) separate the network's prediction space from the loss space and advocate directly predicting clean data x rather than velocity v, while retaining a flow-matching objective in velocity space. Motivated by this distinction, we evaluate the four combinations of x- and v-prediction with x- and v-loss for time-series forecasting. Table 6 reports the corresponding EPF results.

Table 6: Prediction and loss parameterizations on five EPF datasets with known-future covariates and $(W, H) = (168, 24)$. Each column pair specifies the model's direct output (x or v) and the space in which its loss is computed. MSE and MAE are lower-is-better; red boldface marks the best unrounded value for each dataset and metric.

	v-pred/v-loss		x-pred/x-loss		v-pred/x-loss		x-pred/v-loss	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.160**	**0.205**	0.164	0.209	0.161	0.211	0.170	0.213
PJM	0.056	**0.141**	0.057	0.142	0.057	0.143	**0.056**	0.142
BE	0.325	**0.232**	0.325	0.235	**0.319**	0.234	0.361	0.252
FR	**0.269**	**0.171**	0.282	0.172	0.282	0.174	0.289	0.179
DE	**0.208**	**0.286**	0.216	0.292	0.216	0.294	0.230	0.300
AVG	**0.204**	**0.207**	0.209	0.210	0.207	0.211	0.221	0.217

The v-prediction/v-loss configuration yields the lowest five-dataset average MSE and MAE among the four evaluated parameterizations. This differs from the preference for x-prediction on high-dimensional image patches reported by Li & He (2025), suggesting that the preferred parameterization may depend on the data modality.

C.2 Generative Noise Paths

We compare rectified flow, DDPM, and EDM as alternative generative noise paths. Table 7 reports forecasting accuracy for each EPF dataset and on average.

Table 7: Generative noise-path comparison on five EPF datasets with known-future covariates and $(W, H) = (168, 24)$. MSE and MAE are lower-is-better; red boldface marks the best unrounded value for each dataset and metric.

	Rectified flow		DDPM		EDM	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.160**	**0.205**	0.171	0.216	0.221	0.268
PJM	**0.056**	**0.141**	0.059	0.144	0.076	0.178
BE	**0.325**	0.232	0.326	**0.224**	0.431	0.306
FR	**0.269**	0.171	0.277	**0.167**	0.351	0.226
DE	**0.208**	**0.286**	0.243	0.313	0.323	0.377
AVG	**0.204**	**0.207**	0.215	0.213	0.280	0.271

Among the evaluated generative noise paths, rectified flow yields the lowest five-dataset average MSE and MAE under this setup. This observation is consistent with reports from image and video generation (Esser et al., 2024; Chen et al., 2025) and with the motivation for straighter transport paths (Liu et al., 2022a).

C.3 Numerical Solvers

We compare Euler, Heun, and fourth-order Runge–Kutta (RK4) to assess whether a higher-order solver improves forecasting. Table 8 reports their EPF results.

Table 8: Numerical-solver ablation on five EPF datasets with known-future covariates and $(W, H) = (168, 24)$. MSE and MAE are lower-is-better; red boldface marks the best unrounded value for each dataset and metric.

	Euler		Heun		RK4	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.160**	**0.205**	0.167	0.209	0.169	0.211
PJM	**0.056**	**0.141**	0.058	0.145	0.059	0.144
BE	**0.325**	**0.232**	0.326	0.235	0.329	0.236
FR	0.269	**0.171**	**0.268**	0.174	0.269	0.174
DE	**0.208**	**0.286**	0.213	0.290	0.214	0.290
AVG	**0.204**	**0.207**	0.206	0.211	0.208	0.211

Euler achieves the lowest five-dataset average MSE and MAE, and leads on every dataset-level metric except FR MSE, where Heun is slightly lower (0.268 versus 0.269). Heun and RK4 require more model forward passes without improving average forecasting accuracy, so these results favor Euler under the evaluated inference budget.

C.4 Classifier-Free Guidance

We vary the guidance scale and the conditions removed in the guidance branch: target lookback only, covariates only, or both. For conditional velocity $\mathbf{v}_{\text{cond}}$ and the corresponding null-condition velocity $\mathbf{v}_{\text{null}}$ evaluated at the same noisy future state and diffusion time, guidance scale s gives

$$\mathbf{v}_{\text{CFG}} = \mathbf{v}_{\text{null}} + s\,(\mathbf{v}_{\text{cond}} - \mathbf{v}_{\text{null}})\,. \tag{12}$$

In all three variants, each training example drops the specified condition with probability 0.1. The dropped target-history or covariate condition is replaced by zeros after encoding; when both are dropped, one shared Bernoulli decision zeros both representations. The noisy future state and diffusion time are always retained; any condition not selected for dropout is also retained. Figure 10 plots the five-dataset average MSE and CRPS for each choice under the default EPF ablation protocol.

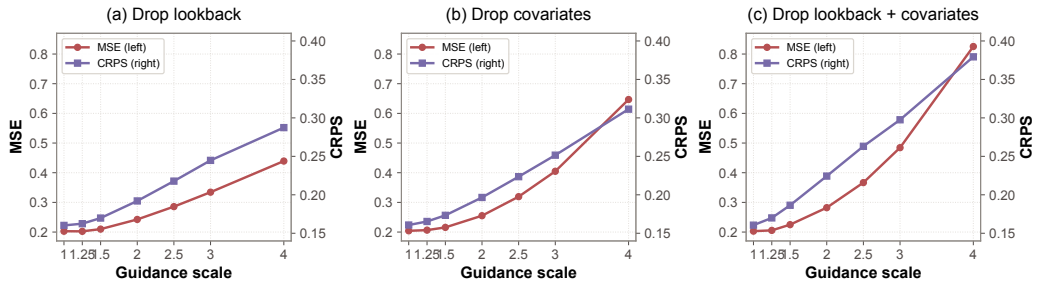


Figure 10: Classifier-free guidance sensitivity on five EPF datasets with known-future covariates and $(W, H) = (168, 24)$. Panels (a)–(c) drop the target lookback, covariates, or both in the guidance branch, respectively. Curves show the arithmetic mean MSE (left axis) and CRPS (right axis) across NP, PJM, BE, FR, and DE; lower is better.

Across all three condition-drop settings, guidance scale 1, which does not amplify the conditional–unconditional difference, gives the most consistent results. Increasing the scale generally degrades both point accuracy and probabilistic forecast quality, especially when both conditions are dropped.

C.5 Hyperparameter Sensitivity

Table 9 reports the complete EPF hyperparameter grid described in Section B, allowing sensitivity to learning rate, model width, and depth to be assessed across datasets.

Table 9: Complete 12-configuration EPF grid with known-future covariates and $(W, H) = (168, 24)$. LR is learning rate, $d = d_{\text{model}}$, $e = e_{\text{layers}}$, and $d_{\text{ff}} = 4d$ throughout. Entries are MSE (lower is better). Red boldface marks the reported best unrounded configuration in each column; values may tie after rounding.

LR	d	e	NP	PJM	BE	FR	DE	AVG
10^{-4}	256	3	0.169	0.057	0.330	0.287	0.248	0.218
3×10^{-4}	256	4	0.166	0.056	0.333	0.280	0.218	0.211
10^{-3}	256	3	0.165	0.056	0.338	0.274	0.220	0.211
10^{-4}	256	4	0.168	0.058	0.333	0.292	0.244	0.219
3×10^{-4}	256	3	0.168	0.056	0.332	0.287	0.218	0.212
10^{-3}	256	4	0.160	0.057	0.335	**0.269**	**0.208**	**0.206**
10^{-4}	512	3	0.165	0.056	**0.325**	0.286	0.235	0.213
3×10^{-4}	512	4	0.161	0.058	0.325	0.277	0.212	0.207
10^{-3}	512	3	0.249	0.060	0.349	0.344	0.216	0.244
10^{-4}	512	4	0.165	**0.056**	0.330	0.289	0.222	0.212
3×10^{-4}	512	3	**0.160**	0.057	0.333	0.284	0.217	0.210
10^{-3}	512	4	0.225	0.060	0.334	0.336	0.215	0.234

D FULL RESULTS

D.1 FLOW MATCHING AND COVARIATE AVAILABILITY

Table 10 reports the per-dataset MSE for flow matching and direct MSE regression under three covariate-availability settings, complementing the aggregate comparison in Figure 5.

Table 10: Flow matching and direct MSE regression on five EPF datasets with $(W, H) = (168, 24)$. The three covariate-availability settings are known-future covariates, past-only covariates, and target history only. The latter two settings are exceptions to the default known-future ablation protocol. Entries are MSE (lower is better); red boldface marks the better unrounded value within each method pair.

	Known-future covariates		Past-only covariates		Target only	
Dataset	Flow Matching	Direct MSE	Flow Matching	Direct MSE	Flow Matching	Direct MSE
NP	**0.160**	0.179	**0.218**	0.241	0.246	**0.240**
PJM	**0.056**	0.070	**0.069**	0.075	**0.077**	0.081
BE	**0.325**	0.326	**0.370**	0.376	0.393	**0.373**
FR	**0.269**	0.320	**0.341**	0.386	**0.362**	0.378
DE	**0.208**	0.242	**0.403**	0.434	0.455	**0.436**
AVG	**0.204**	0.228	**0.280**	0.302	0.307	**0.302**

D.2 FORECASTING WITH EXOGENOUS COVARIATES

We report full deterministic results for EPF (Wang et al., 2024) and probabilistic results for the 13 tasks selected from Shchur et al. (2025).

Experimental Setup and Protocols. For **EPF**, we evaluate $(W, H) = (168, 24)$ both with known-future covariates and with covariates available only in the historical window. We also report a longer EPF prediction horizon, $(W, H) = (720, 360)$, with known-future covariates. Table 12 gives the full known-future results; Table 2 in the main paper gives the $H = 24$ results without future covariates.

For the **13 selected probabilistic tasks**, Table 11 compares DiTS with DeepAR, TimeGrad, TempFlow, and GCGNet using known-future covariates.

We evaluate DiTS under the 12 configurations described in Section B; each baseline is likewise evaluated over 12 configurations per task based on its published settings. For each task, the validation set consists of the windows immediately preceding the test period and matches the test set in window count. To limit computation, we retain at most 100,000 training windows per task, selecting those closest in time to the validation period. We exclude past-only covariates from training inputs for all models because some baselines do not support them.

Following Shchur et al. (2025), we report Scaled Quantile Loss (SQL), Mean Absolute Scaled Error (MASE), Weighted Absolute Percentage Error (WAPE), and Weighted Quantile Loss (WQL). SQL and MASE normalize errors by each series' historical seasonal error, whereas WQL and WAPE aggregate errors in a scale-dependent manner. We report each metric for every task and its geometric mean across the 13 tasks.

Table 11: Probabilistic forecasting on 13 selected tasks with known-future covariates. SQL, MASE, WAPE, and WQL are lower-is-better metrics. Red boldface and blue underlining mark the best and second-best unrounded values in each row, respectively. The geometric mean is calculated separately for each metric over the 13 tasks.

Dataset	Metric	Models				
		DiTS	DeepAR	TimeGrad	TempFlow	GCGNet
ENTSO-e Load (15T)	SQL	0.829	<u>0.801</u>	0.957	0.936	**0.779**
	MASE	0.968	<u>0.945</u>	1.086	1.128	**0.779**
	WAPE	0.077	<u>0.067</u>	0.079	0.078	**0.057**

Table 11: Probabilistic forecasting on 13 selected tasks (continued).

Dataset	Metric	Models				
		DiTS	DeepAR	TimeGrad	TempFlow	GCGNet
	WQL	0.067	0.058	0.070	0.065	**0.057**
ENTSO-e Load (30T)	SQL	0.577	**0.559**	0.668	0.871	0.594
	MASE	0.681	0.684	0.809	1.069	**0.597**
	WAPE	0.055	0.046	0.057	0.077	**0.041**
	WQL	0.047	**0.039**	0.047	0.063	0.041
ENTSO-e Load (1H)	SQL	**0.480**	0.528	0.802	0.525	0.650
	MASE	**0.555**	0.641	0.985	0.645	0.652
	WAPE	**0.037**	0.041	0.064	0.042	0.043
	WQL	**0.032**	0.034	0.052	0.034	0.043
GFC12	SQL	**0.721**	0.828	1.163	0.978	0.940
	MASE	**0.881**	1.015	1.447	1.207	1.046
	WAPE	**0.073**	0.089	0.129	0.108	0.092
	WQL	**0.059**	0.073	0.102	0.086	0.083
GFC14	SQL	**0.471**	0.601	0.683	0.635	1.079
	MASE	**0.564**	0.749	0.819	0.793	1.081
	WAPE	**0.028**	0.037	0.040	0.039	0.053
	WQL	**0.023**	0.030	0.034	0.031	0.053
GFC17	SQL	**0.498**	0.769	0.900	0.940	1.032
	MASE	**0.598**	0.942	1.074	1.142	1.110
	WAPE	**0.039**	0.063	0.069	0.075	0.074
	WQL	**0.032**	0.052	0.058	0.061	0.069
Solar with Weather (15T)	SQL	0.729	**0.710**	0.865	0.873	0.960
	MASE	0.891	**0.871**	1.038	1.050	0.970
	WAPE	0.995	**0.986**	1.219	1.353	1.079
	WQL	0.839	**0.808**	1.017	1.123	1.071
Solar with Weather (1H)	SQL	**0.691**	0.841	0.854	0.888	0.996
	MASE	**0.859**	1.021	1.030	1.066	1.019
	WAPE	**0.909**	1.139	1.036	1.146	1.098
	WQL	**0.782**	0.963	0.855	0.953	1.080
EPF BE	SQL	**0.543**	0.581	0.551	0.543	0.733
	MASE	**0.682**	0.727	0.695	0.697	0.735
	WAPE	**0.123**	0.132	0.124	0.124	0.134
	WQL	0.098	0.105	0.098	**0.096**	0.133
EPF DE	SQL	**0.497**	0.667	0.554	0.529	0.627
	MASE	**0.616**	0.820	0.679	0.684	0.631
	WAPE	**0.264**	0.326	0.437	0.311	0.294
	WQL	**0.213**	0.273	0.348	0.243	0.292
EPF FR	SQL	**0.358**	0.425	0.365	0.394	0.509
	MASE	**0.458**	0.539	0.463	0.495	0.511
	WAPE	**0.068**	0.082	0.069	0.074	0.076
	WQL	**0.054**	0.064	0.054	0.058	0.076
EPF NP	SQL	0.685	0.949	**0.624**	0.659	0.916
	MASE	0.835	1.140	**0.797**	0.855	0.919
	WAPE	0.038	0.051	**0.036**	0.039	0.041
	WQL	0.031	0.042	**0.028**	0.030	0.041
EPF PJM	SQL	0.386	**0.342**	0.374	0.385	0.458
	MASE	0.479	**0.420**	0.465	0.484	0.459
	WAPE	0.084	**0.075**	0.082	0.085	0.080
	WQL	0.068	**0.061**	0.066	0.068	0.079
Geometric Mean	SQL	**0.557**	0.638	0.683	0.671	0.763
	MASE	**0.678**	0.782	0.835	0.833	0.777
	WAPE	**0.098**	0.111	0.123	0.121	0.112
	WQL	**0.081**	0.091	0.100	0.097	0.110

Table 12: Full EPF forecasting results with known-future covariates for $(W, H) = (168, 24)$ and $(720, 360)$. The baselines follow Table 1; GCGNet and DAG results are from Li et al. (2026) and Qiu et al. (2026), respectively. Lower MSE and MAE values are better. AVG matches the two-horizon summary in Table 1. Red boldface and blue underlining mark the best and second-best displayed values in each row and metric.

Models		DiTS		GCGNet		TimeXer		DAG		iTrans.		Cross.		DUET		TFT	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NP	24	**0.160**	**0.205**	0.197	0.233	0.181	0.229	0.202	0.237	0.192	0.250	0.196	0.241	0.246	0.287	0.219	0.249
	360	0.421	0.426	0.496	0.442	0.568	0.482	0.521	0.451	0.516	0.496	**0.404**	**0.406**	0.576	0.528	0.539	0.501
	AVG	**0.290**	**0.315**	0.346	0.337	0.375	0.355	0.362	0.344	0.354	0.373	0.300	0.323	0.411	0.408	0.379	0.375
PJM	24	**0.056**	**0.141**	0.058	0.148	0.079	0.175	0.057	0.143	0.075	0.164	0.087	0.188	0.072	0.166	0.095	0.195
	360	**0.114**	**0.201**	0.128	0.223	0.170	0.275	0.130	0.218	0.120	0.231	0.152	0.236	0.131	0.228	0.133	0.219
	AVG	**0.085**	**0.171**	0.093	0.186	0.124	0.225	0.093	0.180	0.098	0.197	0.119	0.212	0.102	0.197	0.114	0.207
BE	24	0.325	0.232	**0.323**	**0.227**	0.377	0.237	0.361	0.229	0.343	0.250	0.345	0.228	0.432	0.272	0.426	0.272
	360	**0.425**	**0.291**	0.524	0.347	0.486	0.322	0.485	0.330	0.684	0.441	0.461	0.308	0.597	0.436	0.482	0.310
	AVG	**0.375**	**0.261**	0.423	0.287	0.432	0.279	0.423	0.279	0.513	0.345	0.403	0.268	0.515	0.354	0.454	0.291
FR	24	**0.269**	**0.171**	0.332	0.179	0.359	0.193	0.355	**0.171**	0.369	0.203	0.367	0.194	0.384	0.251	0.543	0.253
	360	**0.394**	**0.234**	0.478	0.280	0.471	0.267	0.473	0.268	0.599	0.319	0.457	0.270	0.607	0.403	0.465	0.261
	AVG	**0.332**	**0.202**	0.405	0.230	0.415	0.230	0.414	0.219	0.484	0.261	0.412	0.232	0.496	0.327	0.504	0.257
DE	24	**0.208**	**0.286**	0.282	0.328	0.287	0.328	0.277	0.322	0.261	0.325	0.268	0.323	0.376	0.378	0.380	0.383
	360	**0.331**	**0.348**	0.491	0.445	0.526	0.453	0.462	0.418	0.555	0.488	0.444	0.430	0.589	0.482	0.599	0.509
	AVG	**0.270**	**0.317**	0.387	0.387	0.406	0.391	0.370	0.370	0.408	0.407	0.356	0.376	0.482	0.430	0.489	0.446

D.3 Covariate Conditioning Mechanisms

Table 13 gives the full results for Time-aligned Modulation in DiTS and four alternative conditioning mechanisms: cross-attention, global modulation, joint attention, and variate attention. Figure 6 illustrates these mechanisms.

The four alternatives differ in how covariates enter the target computation. *Cross-attention* lets target tokens query covariate patch embeddings as keys and values through a gated residual branch. *Global modulation* flattens the ordered covariate patch embeddings, maps them to one sequence-level condition, and combines it with diffusion time to generate AdaLN parameters shared across target positions. *Joint attention* concatenates covariate and target tokens so they interact in one attention sequence. *Variate attention* maintains separate target and covariate streams: each first applies attention along time, then joint attention across the target and covariate variates at each temporal position. All four variants inject diffusion time through AdaLN; the diagrams omit this common pathway for clarity.

Table 13: Covariate-conditioning mechanisms on five EPF datasets with known-future covariates and $(W, H) = (168, 24)$. DiTS uses Time-aligned Modulation; the alternatives use cross-attention, global modulation, joint attention, or variate attention. Entries are MSE and MAE (lower is better). Red boldface marks the best unrounded result for each dataset and metric.

Method	NP		PJM		BE		FR		DE		AVG	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DiTS	0.160	**0.205**	**0.056**	**0.141**	**0.325**	0.232	**0.269**	**0.171**	**0.208**	**0.286**	**0.204**	**0.207**
Cross attn.	0.230	0.260	0.072	0.168	0.394	0.261	0.352	0.197	0.397	0.399	0.289	0.257
Global mod.	0.176	0.221	0.058	0.146	0.327	0.240	0.276	0.176	0.262	0.320	0.220	0.221
Joint attn.	0.164	0.210	0.057	0.141	0.334	**0.231**	0.291	0.172	0.230	0.304	0.215	0.212
Variate attn.	**0.158**	0.206	0.057	0.143	0.335	0.238	0.273	0.175	0.213	0.288	0.207	0.210

E SHOWCASES

Figures 11–13 show representative forecasts on EPF (Wang et al., 2024) and the selected probabilistic tasks. The forecast mean and 80% prediction interval are estimated from 10 independently sampled trajectories for EPF and 50 for the selected tasks. Known-future covariates are shown through the prediction horizon, while covariates available only in the past end at the history–forecast boundary.

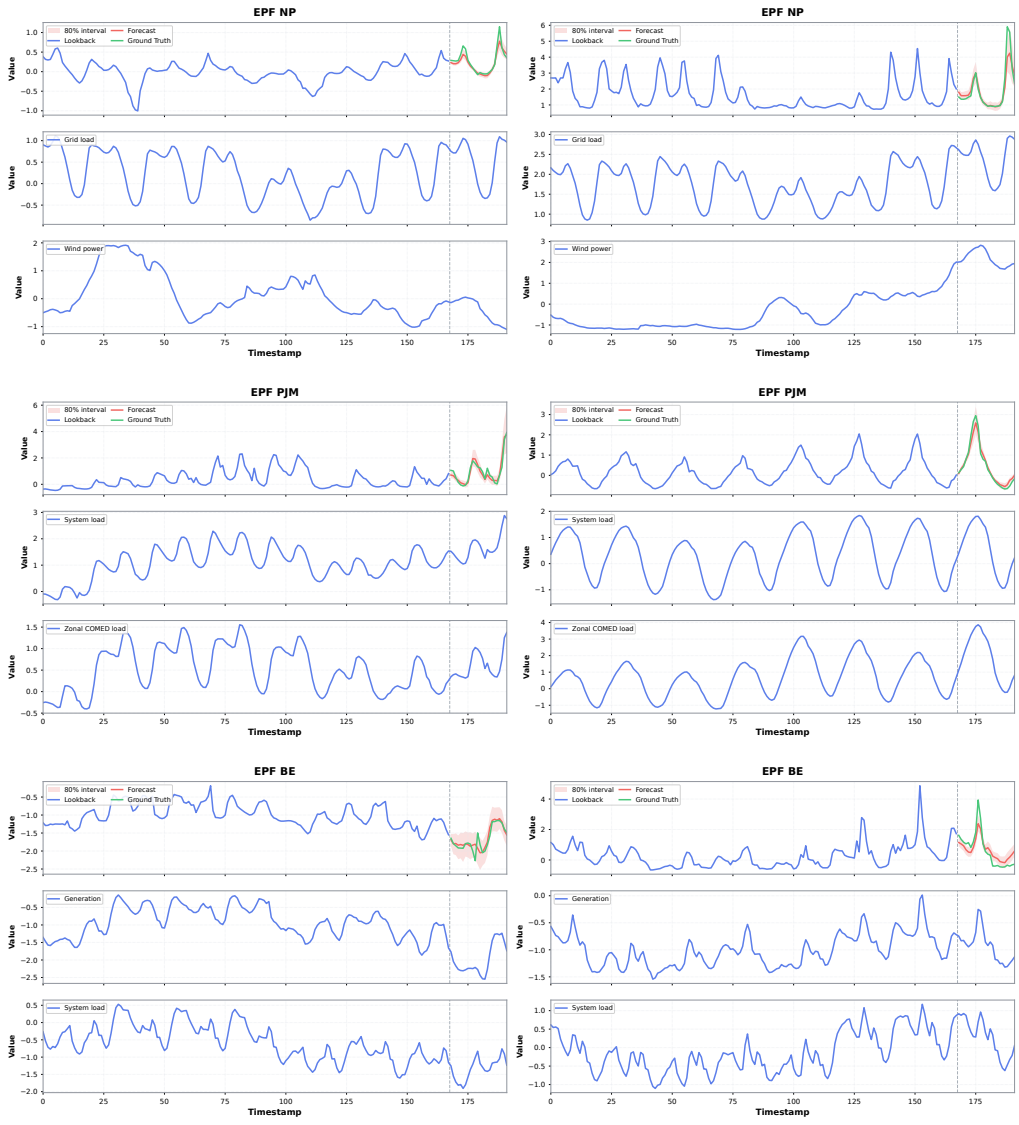


Figure 11: DiTS forecasts on EPF NP, PJM, and BE, with two examples from each dataset. In each example, the top panel shows the target and lower panels show available covariates. The vertical dashed line separates the observed history from the prediction horizon. Blue is the target history, red the mean of 10 sampled forecasts, green the ground truth, and the shaded band the 80% prediction interval. Known-future covariates continue through the prediction horizon.

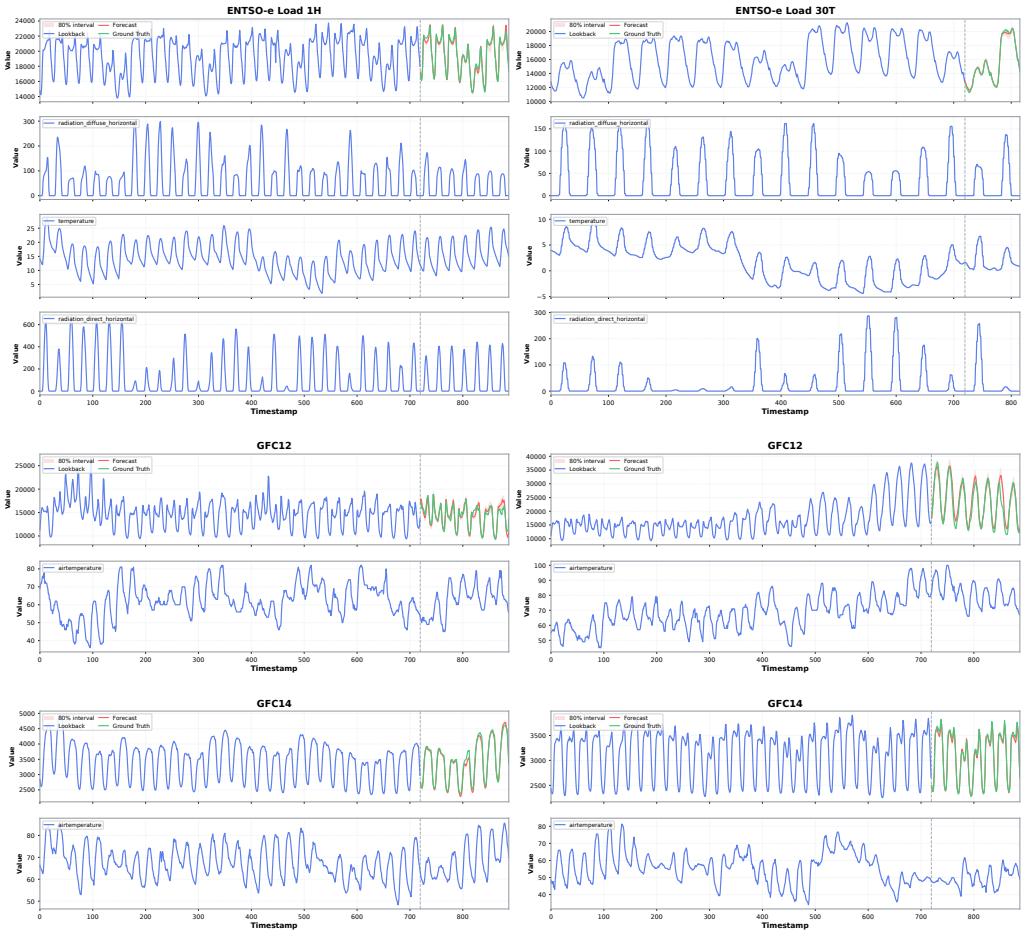


Figure 12: DiTS forecasts on selected electricity-load and GFC tasks: ENTSO-e Load (1H and 30T), GFC12, and GFC14. Each example displays the target above its covariates; the vertical dashed line marks the history–forecast boundary. Blue is the observed target, red the mean of 50 sampled forecasts, green the ground truth, and the shaded band the 80% prediction interval. Covariates extend into the horizon only when known in advance.

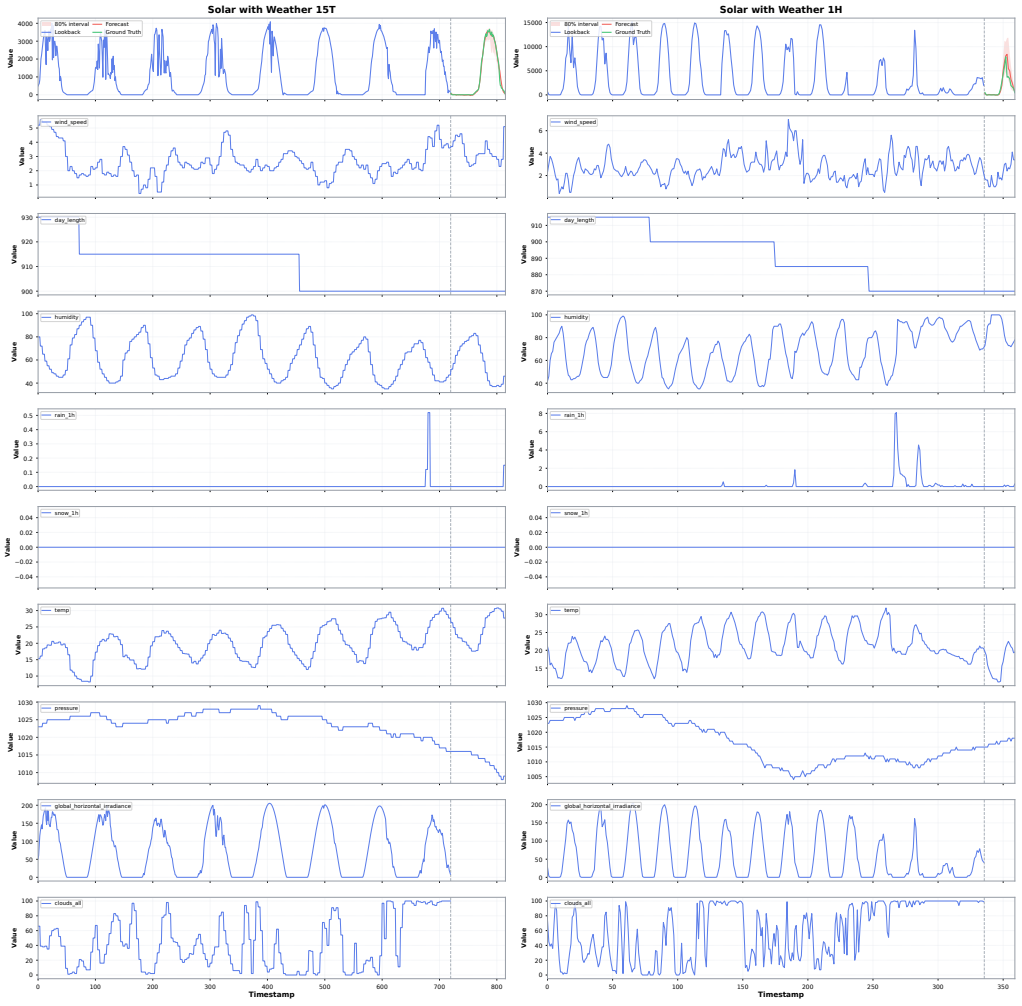


Figure 13: DiTS forecasts on the selected Solar with Weather tasks at 15-minute and hourly resolution. The target appears above the weather covariates, and the vertical dashed line separates history from the forecast horizon. Blue is the observed target, red the mean of 50 sampled forecasts, green the ground truth, and the shaded band the 80% prediction interval. Weather covariates are shown in the horizon when available in advance; past-only covariates stop at the boundary.