Information Bottleneck Learning for Faithful Time Series Forecasting Explanations

Xu Zheng[1], Wei Cheng[2], Zhuomin Chen[1], Mo Sha[1], Jingchao Ni[3], Dongsheng Luo[4]*

[1]Florida International University, Miami, FL, USA
[2]NEC Laboratories America, Princeton, NJ, USA
[3]University of Houston, Houston, TX, USA
[4]Singapore Management University, Singapore
xzhen019@fiu.edu, weicheng@nec-labs.com, zchen051@fiu.edu,
msha@fiu.edu, jni7@uh.edu, luodongsheng01@gmail.com

arXiv:2607.28124v1 [cs.LG] 30 Jul 2026

Abstract

As forecasts increasingly drive decisions in fields such as energy, transportation, and healthcare, understanding the historical data behind these predictions has become as crucial as the predictions themselves. Although existing interpretable-by-design forecasters reveal their internal structures, they offer no guarantee that these structures faithfully reflect the underlying evidence driving the predictions. In contrast, while faithfulness-oriented methods explicitly verify model behavior, they are almost exclusively designed for post-hoc classification tasks. To bridge this gap, we propose IB-Forecast, an inherently interpretable multivariate time-series forecasting framework. It decomposes forecasting into a learned periodic component and a residual component computed with explainable masks over input tokens. With a budget-constrained information bottleneck, end-to-end optimization enables users to directly control explanation sparsity. With a rigorous faithfulness evaluation protocol, extensive experiments demonstrate that IB-Forecast matches the forecasting error of leading black-box models while providing faithful explanations at no additional inference cost. Furthermore, under a matched sparsity budget, these native explanations consistently surpass gradient-based, occlusion-based, and optimization-based baselines across all evaluated datasets. Ultimately, whereas the native explanations of existing interpretable forecasters exhibit poor faithfulness, IB-Forecast guarantees high explanation fidelity, requiring only 14–20% of the observations to deliver low-error predictions.

Introduction

Long-horizon multivariate forecasting increasingly supports decision-making in energy (Zhou et al. 2021), traffic (Li et al. 2018), weather (Rasp et al. 2020), and healthcare (Johnson et al. 2016). Recent advances have produced a rapid succession of increasingly accurate forecasting architectures (Zhou et al. 2021; Nie et al. 2023; Liu et al. 2024a; Lin et al. 2024, 2025). When forecasts drive consequential decisions, understanding which historical inputs influenced a prediction is crucial (Rudin 2019a). However, the most accurate forecasters remain notoriously opaque and lack inherent interpretability (Lim and Zohren 2021).

To unravel the black boxes, two mainstream kinds of methods are introduced. One line of work addresses this problem through intrinsically interpretable forecasting architectures, including the basis decomposition of N-BEATS and N-HiTS (Oreshkin et al. 2020; Challu et al. 2023), the variable-selection networks and interpretable attention mechanisms of the Temporal Fusion Transformer (Lim et al. 2021), prototype-based models (Ming et al. 2019; Peng et al. 2026), and concept-bottleneck approaches (van Sprang, Acar, and Zuidema 2025). These methods are typically evaluated primarily on forecasting error, while the interpretability provided by their internal structures is rarely assessed systematically. Another line of work develops faithfulness-oriented attribution methods that explain a trained predictor after the fact. Recent information-bottleneck approaches (Queen et al. 2023; Liu et al. 2024c; Jang, Kim, and Yang 2025) isolate minimal input subsets that sustain predictive performance, frequently establishing robust evaluation protocols and benchmarks in the process. However, these approaches focuses predominantly on classification and relies on post-hoc explanations for frozen predictors.This decouples the explanation from the actual forecasting computation, frequently requiring the model to evaluate perturbed inputs that fall outside its training distribution (Liu et al. 2024b).

Compared to standard classification, multi-step forecasting requires modeling persistent levels and scales, recurring temporal structures, and input-dependent biases across both variables and future time horizons, making a direct transfer of classification-oriented attribution insufficient. Moreover, a forecast is often determined jointly by regular patterns learned across the training data, such as level, scale, and periodicity, and by instance-specific observations in the current history window (Das et al. 2024; Oreshkin et al. 2020).

To bridge this gap, we introduce IB-Forecast, a self-interpretable multivariate forecasting framework in which the forecast depends on the history window through a disclosed structural context, namely each channel's level and scale together with a learned seasonal profile, and the small set of history tokens selected by a sparse binary mask. Conditioned on this context, the forecast depends only on the emitted mask and the tokens it selects. The explanation is generated within the forward pass and forms part of the forecasting computation itself. To achieve this property, IB-Forecast decomposes each prediction into two readable components by following PatchDecomp (Tomioka and Yoshimura 2026) to

*Corresponding author.

map the input into tokens. First, a learned periodic profile captures recurring temporal patterns, such as daily or weekly variation. Second, a gated deviation readout describes which departures from those recurring patterns are inferred from the current history window. Beyond the disclosed context, all input dependence passes through a budgeted binary mask over patch and channel deviation tokens. Closing a gate prevents its deviation from influencing the readout when the structural context is held fixed.

From the information theoretic perspective, an explanation should preserve the information needed to predict the future while discarding input details that are unnecessary for forecasting (Tian et al. 2020). With this principle, Information Bottleneck(IB) (Tishby, Pereira, and Bialek 2000), we learn the mask by balancing two objectives, retaining sufficient information about the forecasting target and limiting the amount of historical information that can pass through the gated deviation pathway. This balance improves the stability and generalization of both the forecast and its explanation by restricting redundant or unstable details. We further introduce an explicit information budget that allows users to control how much of the historical window the model may consult. Crucially, IB-Forecast guarantees faithfulness by architecturally ensuring that only the tokens selected by the information bottleneck can affect the deviation readout. With a relaxed information budget, IB-Forecast approaches its dense forecasting configuration and can use the complete history window. As the budget becomes tighter, the bottleneck forces the model to prioritize the observations that carry the most forecasting information. This produces a spectrum of models ranging from dense forecasting to highly selective forecasting within a single framework. Across multivariate forecasting benchmarks, IB-Forecast retains competitive predictive accuracy under tight budgets while consulting only a small fraction of the historical input. In summary, we make three main contributions:

- We formulate self-interpretable multivariate forecasting in Problem 1, which asks a single forward pass to jointly produce a forecast and a compact explanation under a predefined budget and a disclosed structural context.
- We propose IB-Forecast, which decomposes each forecast into a learned periodic profile and a gated deviation readout. The gate serves as an information bottleneck, retaining forecasting-relevant information while limiting unnecessary dependence on the historical input.
- Across multivariate benchmarks, IB-Forecast matches strong dense forecasters in accuracy, yields more faithful masks than intrinsic and post-hoc baselines, and outperforms test-time optimization without inference-time optimization. The results also show that accurate forecasts often require only a small fraction of historical input.

Related Work

Interpretable-by-design forecasting. Recent forecasting models achieve interpretability through various structural priors. N-BEATS and NHiTS (Oreshkin et al. 2020; Challu et al. 2023) decompose forecasts into interpretable bases, while DLinear (Zeng et al. 2023) provides transparency in the classical sense through a readable linear map. Other approaches expose specific architectural components to users, such as variable-selection weights and attention in TFT (Lim et al. 2021), per-patch contributions in PatchDecomp (Tomioka and Yoshimura 2026), prototype similarities in ProSeNet and ProtoTS (Ming et al. 2019; Peng et al. 2026), and named concepts in concept-bottleneck transformers (van Sprang, Acar, and Zuidema 2025). However, these methods failed to evaluate exposed structures as faithful explanations.

Post-hoc time-series explanation. Gradient-based techniques (Simonyan, Vedaldi, and Zisserman 2014; Sundararajan, Taly, and Yan 2017; Jang, Kim, and Yang 2025) and perturbation methods dominate post-hoc explanation. A prominent recent line of work focuses on learning per-instance masks at test time, including Dynamask (Crabbé and Van Der Schaar 2021), ExtremalMask (Enguehard 2023), and ContraLSP (Liu et al. 2024c). Information-bottleneck explainers such as TimeX (Queen et al. 2023) and TimeX++ (Liu et al. 2024b) bring information-theoretic rigor and ground-truth benchmarks to this space. Due to designed for classification after training, they can disagree with the computation (Rudin 2019b), which makes robust evaluation an active research area itself (DeYoung et al. 2020; Zheng et al. 2025).

Information bottlenecks and stochastic gates. The information bottleneck principle (Tishby, Pereira, and Bialek 2000) was ported to interpretable graph learning by GSAT (Miao, Liu, and Li 2022), whose stochastic-attention argument, in which injected gate noise penalizes retaining merely correlated inputs, is the blueprint for our gate objective. We use hard binary-Concrete gates (Maddison, Mnih, and Teh 2017; Louizos, Welling, and Kingma 2018) with straight-through estimation (Bengio, Léonard, and Courville 2013).

Preliminaries

Time Series Decomposition. A time series is classically decomposed into three components: a slowly varying *trend*, a recurring *seasonal* pattern, and a *residual* that remains after the first two are removed (Cleveland et al. 1990). For each channel, the decomposition is additive:

$$X[t, c] = T[t, c] + S[t, c] + R[t, c], \tag{1}$$

where T is the trend, S is the seasonal component with period W, such that $S[t, c] = S[t \bmod W, c]$, and R is the residual. By separating stable structures like seasonality from window-specific residuals and local trends, this decomposition isolates what truly drives the forecast. Instead of repeatedly explaining predictable seasonal baselines, the focus is solely on how unique local variations adapt the model to the current context.

We formalize this distinction through a structural context $\mathcal{C} = c(X)$ determined from the input. It records the recurring seasonal profile together with a per-channel frame, consisting of a level μ and scale σ. We assume that, *once $\mathcal{C}$ is fixed, this structural frame spans both the look-back and forecast horizons, extending the seasonal profile periodically*. Consequently, *given $\mathcal{C}$, X and its remainder Δ mutually determine one another, as do the future Y and its remainder Δ_Y*. Motivated by this separation, we define an explanation as compact, window-specific evidence interpreted within the structural

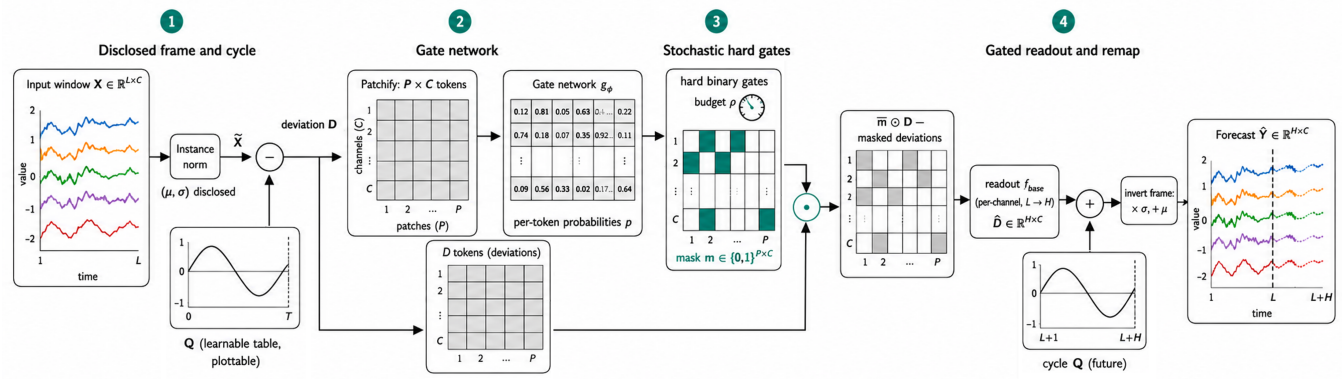

Figure 1: Overview of IB-Forecast. Stage 1 normalizes the input X and removes the learned periodic profile Q to obtain the deviation D. Stage 2 partitions D into tokens and scores each with a small transformer gate network. Stage 3 samples a hard binary gate for each token to form the mask m. Stage 4 computes the forecast from the masked deviations $\bar{m} \odot D$ and remaps the output back to the original space. The mask m, profile Q, and normalization information constitute the explanation E.

context. In terms of Eq. (1), the explanation concerns only the residual R and the local variation of the trend T.

Forecasting & Explanation. Let $X \in \mathbb{R}^{L\times C}$ denote a look-back window of C channels over L time steps, where $X[t, c]$ is the value of channel c at time step t. We use X to denote the input window, from which the context $\mathcal{C} = c(X)$ is constructed. The training set $\mathcal{T} = \{(X_i, Y_i) \mid i \in [N]\}$ contains input windows X_i and targets $Y_i \in \mathbb{R}^{H\times C}$, where each target contains the next H observations of all channels. A conventional forecaster is a function $f : \mathcal{X} \to \mathbb{R}^{H\times C}$ learned by minimizing $\mathbb{E}\|f(X) - Y\|_2^2$.

Post-hoc attribution explains a frozen forecaster after training, but it does not guarantee that the forecast was computed through the resulting explanation. We instead require the forecast to be recoverable from the explanation itself. Inspired by the information bottleneck principle (Tishby, Pereira, and Bialek 2000), we formulate the following problem.

Problem 1 (Self-interpretable forecasting)**.** *Learn a forecaster that, in a single forward pass, emits a forecast and an explanation, $(\hat{Y}, E) = f(X)$, while disclosing a structural context $\mathcal{C}$, such as a seasonal pattern and its associated per-channel frame. The explanation must satisfy two properties:*

1. ***Compactness**: the attribution E selects only a small, user-controlled fraction of the input.*
2. ***Conditional faithfulness by construction**: conditioned on $\mathcal{C}$, the forecast depends only on E; that is, $\hat{Y} = g(E; \mathcal{C})$ with the readout function g.*

Thus, once the structural context is fixed, any window-specific information excluded from E cannot affect the forecast.

The explanation E is generated solely from the observed window. This prevents explanations from peeking at the forecasting horizon. Conditioned on $\mathcal{C}$, *the construction enforces the Markov chain $X \to E \to \hat{Y} \mid C$*, so that E is the only carrier of window-specific predictive information. In terms of Eq. (1), *E therefore selects evidence about the residual R and local changes in the trend T*, while the seasonal component S belongs to $\mathcal{C}$ and is never included in the attribution. The conditioning applies symmetrically to the forecasting. *Predicting Y given $\mathcal{C}$ is therefore equivalent to predicting its remainder Δ_Y*. Accordingly, the explanation should be evaluated on the deseasonalized and normalized future.

Methodology

In this section, we introduce IB-Forecast, an interpretable framework for multivariate time-series forecasting whose explanation is generated within the same forward pass and directly determines the forecast. Figure 1 illustrates the framework, which implements the decomposition in Eq. (1) under the formulation in Problem 1. The framework consists of four stages. The first stage isolates the disclosed structural context $\mathcal{C}$ by normalizing each channel to remove its level and scale and then subtracting a learned seasonal profile, leaving only the window-specific deviations. The second and third stages construct a stochastic binary mask that selects a small, user-specified budget of deviation tokens defined over patches and channels. The fourth stage computes the deviation forecast solely from the selected tokens, after which the seasonal profile, level, and scale are restored. A mask applied directly to the raw window would use the budget in property (i) to retransmit the level and recurring rhythm, while setting a closed token to the implausible raw value of zero. In contrast, closing a deviation token expresses the in-distribution assumption that the corresponding observation follows the seasonal pattern at the current window level. Consequently, the selected deviations form the only window-specific pathway to the output. Given the disclosed context $\mathcal{C}$, the mask itself serves as the explanation, and every excluded token is structurally prevented from influencing the forecast. The remainder of this section defines the explanation object in Problem 1, presents the four-stage architecture and training objective, and explains how the final explanation is obtained.

Normalization and seasonal profile. The structural context is removed in two steps. First, each channel is normalized at the instance level, removing the window's own per-channel mean and standard deviation $\mu, \sigma \in \mathbb{R}^{1\times C}$ (Kim et al. 2022). Second, the recurring seasonal pattern is subtracted (Lin et al.

2024). It is stored as a lookup table $Q \in \mathbb{R}^{W \times C}$ of learnable parameters covering one full cycle of length W, fixed from the sampling rate (e.g., $W{=}24$ on hourly data with a daily cycle). Entry $Q[w, c]$ is the typical normalized value of channel c at phase w of the cycle, learned jointly with the rest of the model, so a column of Q plots as, for instance, the average daily load curve of that channel. With τ the window's phase within the cycle, known from its timestamp,

$$\tilde{X} = \frac{X - \mu}{\sigma}, \qquad D = \tilde{X} - Q^{\mathrm{per}}_{[\tau:\tau+L]}, \tag{2}$$

where $Q^{\mathrm{per}}[t, c] = Q[t \bmod W, c]$, which lets one cycle cover a look-back and horizon of any length, so when $L > W$ the profile simply tiles across the window. The subtraction leaves in the deviation D only what is specific to this window, the remaining trend movements and residual structure, which the readout forecasts jointly.

Amortized gate network. The mask operates on (patch × channel) deviation tokens. We split D into $P = L/L_p$ patches per channel. For patch p and channel c, the token is

$$d_{p,c} = \big(D[(p{-}1)L_p{+}1, c], \ldots, D[pL_p, c]\big) \in \mathbb{R}^{L_p}, \tag{3}$$

which contains the L_p consecutive deviation values in that patch. Each token is mapped to a d-dimensional representation using a shared linear embedding $W_e \in \mathbb{R}^{d \times L_p}$ and a learned patch-position embedding $e_p \in \mathbb{R}^d$,

$$v_{p,c} = W_e d_{p,c} + e_p. \tag{4}$$

The gate network then scores all tokens. A transformer encoder g_ϕ contextualizes the token embeddings, and a linear head $W_g \in \mathbb{R}^{1 \times d}$ produces a logit and opening probability for each token,

$$p_{p,c} = \mathrm{sig}(W_g g_\phi(v)_{p,c}), \tag{5}$$

where $\mathrm{sig}(\cdot)$ denotes the logistic sigmoid. Since each score is computed from a contextualized representation, the gate network learns to compare tokens within the window and rank them by their forecasting relevance. This amortized scoring makes the explanation available in a single forward pass, avoiding per-instance test-time mask optimization. For datasets with many channels, we apply the encoder within each channel over its P patches, rather than over all PC tokens jointly. This reduces the attention cost from $O((PC)^2)$ to $O(CP^2)$ while keeping the gating procedure tractable for high-dimensional multivariate series.

Stochastic hard gates. To ensure that a closed token is truly dropped, we adopt binary gates. The challenge lies in optimization, since discrete gates carry no gradient. We sample each gate during training from a noisy relaxation of its logit and binarize it with a straight-through estimator (Louizos, Welling, and Kingma 2018). Formally, we have

$$m_{p,c} = \mathbb{1}\big[b_{p,c} > \tfrac{1}{2}\big], b_{p,c} = \mathrm{sig}\Big(\tfrac{\ell_{p,c}+\varepsilon}{T}\Big), \tag{6}$$

where $\varepsilon = \log u - \log(1-u), u \sim \mathrm{Uniform}(0,1)$, the forward pass uses the binary $m_{p,c}$, the backward pass differentiates through the smooth $b_{p,c}$, and the temperature T is annealed over training. During inference, the mask is deterministic and noise-free.

Algorithm 1 IB-Forecast forward pass

Require: window $X \in \mathbb{R}^{L \times C}$, phase τ
Ensure: forecast $\hat{Y}$, explanation E

$\mu, \sigma \leftarrow$ per-channel mean and std. of X
$D \leftarrow (X - \mu)/\sigma - Q^{\mathrm{per}}_{[\tau:\tau+L]}$ — Eq. (2)
$v_{p,c} \leftarrow W_e\, d_{p,c} + e_p$ for all (p, c) — Eqs. (3)–(4)
$p_{p,c} \leftarrow \mathrm{sig}\big(W_g\, g_\phi(v)_{p,c}\big)$ — Eq. (5)
if training **then**
  $m \leftarrow$ hard-Concrete sample — Eq. (6)
else
  $m \leftarrow \mathbb{I}\big[p > \frac{1}{2}\big]$
end if
$\hat{D} \leftarrow f_{\text{base}}(\bar{m} \odot D)$ — Eq. (7)
$\hat{Y} \leftarrow \sigma \odot \big(\hat{D} + Q^{\mathrm{per}}_{[\tau+L:\tau+L+H]}\big) + \mu$ — Eq. (8)
return $\hat{Y}$ and $E = \big(m,\ \bar{m} \odot D\big)$, with context $C = (\tau;\ \mu, \sigma;\ Q)$

Gated deviation readout. The readout predicts the future deviations $\hat{D} \in \mathbb{R}^{H \times C}$ from the masked deviations alone,

$$\hat{D} = f_{\text{base}}\big(\bar{m} \odot D\big), \tag{7}$$

where $\bar{m} = m \otimes \mathbf{1}L_p$ broadcasts each gate across the L_p time steps in its patch. The function $f$base is a shared channel-wise feed-forward network that maps each masked deviation sequence from L historical steps to H future steps, jointly capturing future residual variation and local trend changes. The final forecast re-applies the structural context, adding the future slice of the seasonal profile and inverting the frame,

$$\hat{Y} = \sigma \odot \big(\hat{D} + Q^{\mathrm{per}}_{[\tau+L:\tau+L+H]}\big) + \mu. \tag{8}$$

When a token is closed, it contributes zero in deviation space. After the seasonal profile, scale, and level are restored, this zero deviation corresponds to the value $\mu + \sigma \odot Q$, meaning that the token follows the expected seasonal pattern at the current window level. The same interpretation is used in our evaluation of faithfulness. Deleting a token resets it to the seasonal profile while keeping the structural context fixed, so the architecture and evaluation metric share the same definition of an absent input.

Training objective All parameters, comprising the profile Q, the gate network, and the readout, are trained end to end by minimizing the formulation:

$$\begin{aligned}\mathcal{L} = \big\|\hat{Y} - Y\big\|_2^2 \ +\ \beta \sum_{p,c} \mathrm{KL}\big(\mathrm{Bern}(p_{p,c}) \,\|\, \mathrm{Bern}(\pi)\big) \\ +\ \lambda_b\big(\overline{p} - \rho\big)^2 \ +\ \lambda_{\text{tv}}\,\mathrm{TV}(m),\end{aligned} \tag{9}$$

where π is a prior open probability, $\overline{p}$ is the mean gate probability over all tokens, $\rho \in (0, 1]$ is the user-chosen budget of Problem 1, and β, λ_b, λ_{tv} are weighting coefficients. The first term is the forecasting loss. The second term regularizes each gate toward a sparse Bernoulli prior (Miao, Liu, and Li 2022). The last term, $\mathrm{TV}(m) = \sum_{p,c} |m_{p+1,c} - m_{p,c}|$, encourages temporally contiguous selections.

Explanation extraction Algorithm 1 summarizes the forward pass, which emits the forecast and the explanation together. The explanation costs no additional computation compared to other methods, since every component of E is the output during computing $\hat{Y}$, and one forward pass amounts to one gate-network encoding and one readout application. When a ranking over tokens is needed, the graded probabilities p are emitted alongside E and provide it, and point saliency for comparison with point-level explainers is p broadcast over each patch's L_p steps.

Theoretical Analysis

Throughout the statement in previous section, the context $\mathcal{C} = (\tau;\, \mu, \sigma;\, Q)$ is computed from the window and its timestamp, Eq. (2) is invertible once (μ, σ, Q, τ) are fixed, so $\Delta = D$, and the stochastic gates of Eq. (6) use independent noise. The conditional information bottleneck is

$$\min_{E} \; -I(E; Y \mid \mathcal{C}) + \beta I(X; E \mid \mathcal{C}), \tag{10}$$

where $I(E; Y \mid \mathcal{C})$ is the explanation's relevance, and $I(X; E \mid \mathcal{C})$ is rate, the information retained from X. Conditioning the bottleneck on $\mathcal{C}$ is what keeps the explanation from being calculated twice for globally predictable structure. Since $\mathcal{C} = (\tau; \mu, \sigma; Q)$ is disclosed, the saved rate is exactly $I(\mathcal{C}; E)$: the explanation is not penalized for information already provided by the context. By contrast, any ε-sufficient unconditional explanation must spend at least $-I(\mathcal{C}; Y) - \varepsilon$ bits recovering the level, scale, and cycle. We formalize this in Theorem 3 in the Appendix.

Theorem 1 (Relevance controls the error floor)**.** *Every forecaster of the form $\hat{Y} = g(E; \mathcal{C})$ satisfies*

$$\mathbb{E}\|Y - \hat{Y}\|_2^2 \geq \frac{n}{2\pi e} \exp\Big(\frac{2}{n}\big(h(Y \mid \mathcal{C}) - I(E; Y \mid \mathcal{C})\big)\Big),$$

where $h(Y \mid \mathcal{C})$ is the conditional differential entropy of the future given the context. The bound decreases in the relevance $I(E; Y \mid \mathcal{C})$. For a fixed set of opened token positions $\mathcal{M} \subseteq \{1, \ldots, P\} \times \{1, \ldots, C\}$, the support of the mask, write $E_{\mathcal{M}} = (\mathcal{M}, \Delta_{\mathcal{M}})$ for the selective explanation that discloses the remainder values at exactly those positions. Then for nested selections $\mathcal{M} \subseteq \mathcal{M}'$,

$$\mathbb{E}\|Y - \mathbb{E}[Y \mid E_{\mathcal{M}'}, \mathcal{C}]\|_2^2 \leq \mathbb{E}\|Y - \mathbb{E}[Y \mid E_{\mathcal{M}}, \mathcal{C}]\|_2^2,$$

so the best achievable Bayes error is nonincreasing as the mask budget grows.

This theorem links explanation quality directly to forecasting accuracy. Once the context $\mathcal{C}$ is set, the forecast is only as good as the information contained in E. Higher relevance, $I(E; Y \mid \mathcal{C})$, drives down the minimum possible error. Naturally, allowing the model to see more tokens can only improve its best possible prediction, creating a clear trade-off between sparsity and accuracy controlled by β or ρ. Due to the page limitations, we provide the detailed proofs in the Appendix.

Experiments

Datasets. We use standard multivariate benchmarks, **ETTh1**, **ETTh2**, **ETTm1**, **ETTm2** (Zhou et al. 2021), **Weather**, and **Electricity**, spanning 7 to 21 channels; dataset statistics and splits are in Appendix . We adopt the evaluation protocol of TQNet (Lin et al. 2025), with look-back 96, horizons $\{96, 192, 336, 720\}$, and early stopping and model selection on the validation split only.

Baselines. The baselines cover both forecasting and explanation. For forecasting, we compare against two groups. (i). *Self-interpretable forecasters*, including **TFT** (Lim et al. 2021), **PatchDecomp** (Tomioka and Yoshimura 2026), **DLinear** (Zeng et al. 2023), and **ProtoTS** (Peng et al. 2026). (ii). *Black-box state-of-the-art forecasters*, namely **TQNet** (Lin et al. 2025), **CycleNet** (Lin et al. 2024), **iTransformer** (Liu et al. 2024a), **PatchTST** (Nie et al. 2023), and **TimesNet** (Wu et al. 2023). For explanation, each self-interpretable forecaster is scored with its own native explanations, and we additionally compare against post-hoc explainers applied to our own trained model, including **gradient-based saliency** (Simonyan, Vedaldi, and Zisserman 2014), **Integrated Gradients (IG)** Sundararajan, Taly, and Yan 2017, and **TIMING** (Jang, Kim, and Yang 2025). We further include perturbation-based patch-level occlusion, a uniform-random baseline, and two test-time mask optimization methods, **DynaMask** (Crabbé and Van Der Schaar 2021) and **ExtremalMask** (Enguehard 2023). Detailed descriptions, implementation adaptations, and complete hyperparameter configurations for all baselines are provided in the Appendix.

Metrics. We follow previous paper (Lin et al. 2025) and use **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** to measure the forecasting accuracy. We measure faithfulness using matched-budget fidelity (ERASER; DeYoung et al. 2020). This relies on two complementary metrics based on forecast shifts: **comprehensiveness@**k (deleting the top-k inputs, where higher is better) and **sufficiency@**k (keeping only the top-k inputs, where lower is better). After normalizing both shifts against an all-masked baseline, the final fidelity score is simply comprehensiveness@k minus sufficiency@k. To ensure fair and consistent evaluation, we apply three corrections to all explainers: (i) we fix k per sample to the explainer's native open count to avoid scoring unselected regions; (ii) we freeze instance-normalization stats (μ, σ) at their original values, treating them as disclosed structure; and (iii) we impute deleted values using the model's own reference semantics ($\mu + \sigma\, Q(t)$ or the window mean), ensuring all perturbations remain in-distribution.

Forecasting Accuracy. As Table 1 lists, IB-Forecast is the best interpretable forecaster on every dataset in MSE and on all but ETTm1 in MAE. PatchDecomp is the closest competitor, trailing by about 4% in overall MSE: it is essentially tied on ETTm1 but loses ground everywhere else. DLinear trails by 18% overall; its single linear map holds up on Weather but degrades sharply on ETTh2 and ETTm2, by 47% and 29%, where a fixed linear read-out cannot adapt to the series. TFT trails on every dataset and by 22% overall, so attention-based transparency buys no accuracy here. ProtoTS stays close only on ETTh2 and falls 37 to 48% behind on the other ETT datasets. The pattern is consistent: each incumbent pays for its transparency with a capacity restriction that at least one

Table 1: Forecasting accuracy on average. Bold is best overall, and underline is best interpretable model. Per-horizon results in the Appendix. ProtoTS in Weather cells exceeded the compute budget (10 hours) and are omitted.

	Interpretable-by-design										*Black-box*							
	IB-Forecast		PatchDecomp		TFT		DLinear		ProtoTS[§]		TQNet		CycleNet		iTransformer		PatchTST	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.438**	**0.429**	0.456	0.446	0.581	0.514	0.456	0.452	0.601	0.521	0.441	0.434	0.457	0.441	0.454	0.448	0.469	0.455
ETTh2	0.381	**0.402**	0.399	0.420	0.448	0.444	0.559	0.515	0.402	0.418	**0.378**	**0.402**	0.388	0.409	0.383	0.407	0.387	0.407
ETTm1	0.383	0.403	0.384	0.399	0.487	0.469	0.403	0.407	0.548	0.473	**0.377**	**0.393**	0.379	0.396	0.407	0.410	0.387	0.400
ETTm2	0.272	0.316	0.297	0.344	0.311	0.345	0.350	0.401	0.403	0.384	0.277	0.323	**0.266**	**0.314**	0.288	0.332	0.281	0.326
Weather	0.244	0.271	0.253	0.276	0.275	0.298	0.265	0.317	—	—	**0.242**	**0.269**	0.243	0.271	0.258	0.278	0.259	0.273
Overall	0.344	**0.364**	0.358	0.377	0.420	0.414	0.407	0.418	—	—	**0.343**	**0.364**	0.347	0.366	0.358	0.375	0.356	0.372

Table 2: Matched-budget fidelity (ρ=0.2, H=96, 1024 test windows): Comp↑, Suff↓, Score = Comp − Suff, with budget k fixed per sample to the native mask's open count. Bold: best per row. [†]Oracle-tuned (Appendix). [‡]Explains its own model's forecast.

		Post-hoc							*Native*			
Dataset	Metric	Occlusion	TIMING[†]	IG	Saliency	Dynamask	ExtremalMask	Random	DLinear[‡]	PatchDecomp[‡]	TFT[‡]	**IB-Forecast**
ETTh1	Comp↑	0.991	0.939	0.922	0.917	0.730	0.740	0.261	0.588	0.186	0.798	**1.019**
	Suff↓	0.171	0.279	0.356	0.335	0.420	0.449	0.943	0.201	0.398	2.276	**0.069**
	Score↑	0.820	0.660	0.567	0.581	0.310	0.292	−0.682	0.387	−0.211	−1.478	**0.950**
ETTm1	Comp↑	0.940	0.947	0.942	0.889	0.751	0.711	0.357	0.695	0.303	0.565	**1.056**
	Suff↓	0.303	0.393	0.396	0.429	0.535	0.563	0.904	0.174	0.468	0.912	**0.133**
	Score↑	0.637	0.555	0.547	0.460	0.216	0.148	−0.547	0.521	−0.165	−0.347	**0.923**
Weather	Comp↑	0.819	0.863	0.859	0.797	0.623	0.561	0.118	0.879	0.726	0.646	**0.925**
	Suff↓	0.116	0.092	0.091	0.131	0.235	0.301	0.709	**0.042**	0.318	0.226	0.077
	Score↑	0.703	0.771	0.768	0.665	0.389	0.260	−0.591	0.838	0.408	0.420	**0.848**

dataset exposes, whereas IB-Forecast is the only model in the group whose accuracy stays on the state-of-the-art frontier across all five. Detailed per-horizon results and analysis, including PatchDecomp's short- versus long-horizon split and ProtoTS's long-horizon collapse, are given in Appendix.

Comparing against black-box state-of-the-art forecasters, in overall MSE, IB-Forecast is statistically tied with TQNet (Δ=0.2%, within seed noise) and ahead of CycleNet, PatchTST, and iTransformer. In overall MAE, it shares the best score with TQNet exactly. At the dataset level, it is the best model outright on ETTh1 and stays within 2.5% of the best black-box model everywhere else, so no dataset exposes an interpretability tax. In the Appendix, we provide per-horizon numbers, seed variability, configuration details.

Faithfulness under matched-budget fidelity. In Table 2, we report matched-budget fidelity on ETTh1, ETTm1, and Weather, at open rates of 13.9%, 16.8%, and 20.4%, respectively. The proposed native mask achieves the highest Score on all three datasets. It combines high comprehensiveness with low sufficiency, showing that the selected inputs both capture the forecast shift and preserve the prediction.

Among post-hoc explainers, Occlusion is the strongest baseline, but it still trails the native mask by 0.13 to 0.29 in Score. Compared with IG and Saliency, it suggests that its estimator helps within the gradient-based family. Dynamask and ExtremalMask deserve a closer look. Despite 500 optimization steps per test batch and the same or an oracle budget, their sufficiency ranges from 0.24 to 0.56 across the three datasets against 0.07 to 0.13 for the native mask, and their best Score is 0.39 against the native 0.85 to 0.95. This indicates that per-instance search cannot recover the sparse computation installed by end-to-end training.

The native baselines show that architectural transparency alone is not enough. TFT and PatchDecomp obtain negative Scores on two of the three datasets, meaning that keeping their top credited inputs often fails to reproduce their own forecasts. DLinear provides a complementary case. Its attribution is mathematically exact, since it is computed from the linear weights in closed form, and its accuracy matches the published DLinear results within 2%. But at the same budget, scores vary from 0.39 to 0.84, indicating influence is spread across the full window rather than concentrated in a sparse subset.

Accuracy and explanation trade-off. To measure this tradeoff, we sweep the budget ρ from 0.05 to 0.5, retrain one model at each setting, and plot test MSE and MAE against the achieved budget in Figure 2. At the budgets used in our main experiments, the mask reads only 13 to 20% of the input, yet the accuracy loss remains small. The MSE gap from the full input baseline is at most 3.4% on ETTh1, 3.5% on ETTm1, and 2% on Weather, while MAE stays within about 6%. The gap shrinks at longer horizons and with looser budgets.

The frontier is different across datasets. On ETTm1, as the allowed read rate increases, both MSE and MAE improve smoothly, from about $+6\%$ error at a 3% read to roughly

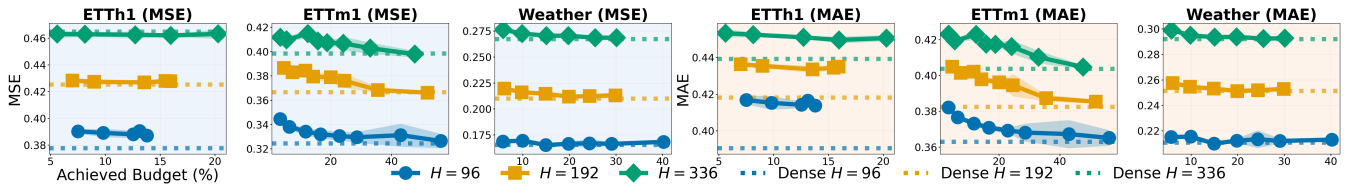

Figure 2: Trade-off between explanation sparsity and forecasting error. Test MSE (left) and MAE (right) against the measured fraction of the input read, per dataset, for horizons $H \in \{96, 192, 336\}$; dotted lines mark the same-horizon dense reference. Points are seed means of three seeds with a standard deviation band.

Table 3: Ablation study on ETTh1. *Open* is the realized base-mask open rate. ‡Fidelity is vacuous when the mask saturates fully open (100%), as it then excludes nothing.

	MSE↓	MAE↓	Open	Comp↑	Suff↓	Score↑
IB-Forecast	0.388	0.411	0.136	1.030	0.071	0.959
Modules						
− gated base	0.382	0.394	0.000	0.000	1.000	−1.000
− hard gates	0.378	0.407	0.099	0.870	0.249	0.621
− cycle Q	0.428	0.430	0.242	1.009	0.011	0.999
Loss terms						
− budget	0.380	0.392	1.000	1.000	0.000	1.000‡
− gate TV	0.387	0.411	0.135	1.028	0.071	0.957
− diversity	0.391	0.412	0.145	1.058	0.096	0.961

the full input baseline near a 50% read. ETTh1 and Weather behave differently. Their unconstrained models already ignore much of the window, using about 20% of the input on ETTh1 and 40 to 50% on Weather. MSE and MAE also reveal slightly different behavior. For instance, on ETTh1, MSE suggests that sparsity is nearly free, including a 0.4 to 0.6% gain at H=336, while MAE still shows a 3 to 6% penalty. Overall, sparse forecasting preserves accuracy with little differences while making the model's input dependence explicit.

Case study. Figure 3 shows a complete explanation for one ETTh1 test window from ρ=0.2. Each row represents one channel, so all model quantities are aligned on a shared time axis. For this example, the model opens only 16% of the (patch $\times$ channel) tokens. These open gates are the only window-specific values the deviation readout is allowed to use, so the highlighted regions are not a post-hoc saliency estimate but the actual evidence used by the forecast (Problem 1). Channels whose behavior is already explained by the periodic profile, the current level, and the current scale can remain fully closed. The open gates, in turn, mark where the window deviates from this disclosed structure. These masked deviations $\bar{m} \odot D$ are all that the readout of Eq. (7) consumes.

Ablation study In Table 3, we conduct ablation studies on the architectural modules and the loss terms on ETTh1 at H=96. As the results show, removing the gated base or the hard gates barely moves MSE and MAE, which both stay within 2.5% of the full model, yet either one destroys the explanation. Without gating, the mask receives no gradient

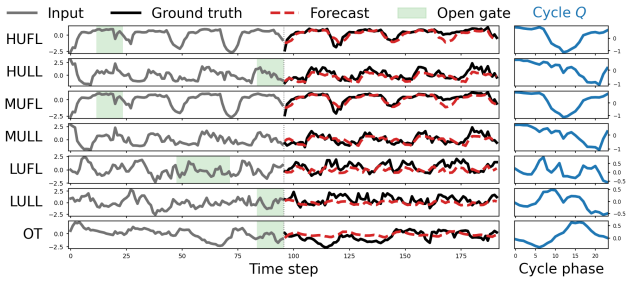

Figure 3: An example of IB-Forecast explanation for one ETTh1 test window (ρ=0.2), one row per channel in RevIN-normalized space. Each row aligns, on a shared time axis, the lookback input (gray), the ground truth (black) and forecast (red dashed), with the mask's open gates (shaded green). The narrow right panel is the learned periodic profile Q.

from the forecast and collapses fully closed (0% open, score −1.0), so the reported saliency is empty. With soft gates the mask leaks a fraction of every token, so keeping its top-k no longer reproduces the forecast and fidelity falls from 0.96 to 0.62. Dropping the budget term leaves accuracy untouched but saturates the mask fully open (100%); its fidelity score of 1.0 is vacuous, since a mask over the entire input excludes nothing and matched-budget fidelity is then trivially maximal. The gated base and the budget are complementary and together they force the mask to be a selective subset. Removing the gate-TV or the factor-diversity penalty has little influence on fidelity. The recurrent cycle Q is the most important part for forecasting. Removing it will cause 10% MSE performance drop. Beyond these on/off ablations, we also provide detailed hyperparameter analysis in the Appendix.

Conclusion

We presented IB-Forecast, a multivariate forecaster with faithful explanations at no accuracy cost. Its budgeted stochastic mask selects deviation tokens, while disclosed level, scale, and seasonal context reconstruct the forecast; thus, the selected tokens are the only window-specific input pathway. IB-Forecast matches state-of-the-art black-box accuracy, achieves stronger matched-budget fidelity than existing interpretable forecasters, and outperforms test-time optimizers on their own objective at zero inference cost. Together, certified explanations and matched-budget evaluation provide a rigorous foundation for interpretable forecasting.

References

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. arXiv:1308.3432.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Ramirez, F. G.; Canseco, M. M.; and Dubrawski, A. 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 6989–6997.

Cleveland, R. B.; Cleveland, W. S.; McRae, J. E.; and Terpenning, I. 1990. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, 6(1): 3–73.

Crabbé, J.; and Van Der Schaar, M. 2021. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, 2166–2177. PMLR.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 10148–10167. PMLR.

DeYoung, J.; Jain, S.; Rajani, N. F.; Lehman, E.; Xiong, C.; Socher, R.; and Wallace, B. C. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Association for Computational Linguistics*, 4443–4458. Association for Computational Linguistics.

Enguehard, J. 2023. Learning Perturbations to Explain Time Series Predictions. In *International Conference on Machine Learning*, volume 202, 9329–9342. PMLR.

Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 3543–3556. Association for Computational Linguistics.

Jang, H.; Kim, C.; and Yang, E. 2025. TIMING: Temporality-Aware Integrated Gradients for Time Series Explanation. In *International Conference on Machine Learning*, 26877–26895. PMLR.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1): 1–9.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2022. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

Lim, B.; Arık, S. O.; Loeff, N.; and Pfister, T. 2021. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.

Lim, B.; and Zohren, S. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194): 20200209.

Lin, S.; Chen, H.; Wu, H.; Qiu, C.; and Lin, W. 2025. Temporal Query Network for Efficient Multivariate Time Series Forecasting. In *International Conference on Machine Learning*, volume 267, 37797–37814. PMLR.

Lin, S.; Lin, W.; Hu, X.; Wu, W.; Mo, R.; and Zhong, H. 2024. CycleNet: Enhancing Time Series Forecasting through Modeling Periodic Patterns. In *Advances in Neural Information Processing Systems*, volume 37, 106315–106345. Curran Associates, Inc.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024a. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*, volume 2024, 11116–11140.

Liu, Z.; Wang, T.; Shi, J.; Zheng, X.; Chen, Z.; Song, L.; Dong, W.; Obeysekera, J.; Shirani, F.; and Luo, D. 2024b. TimeX++: Learning Time-Series Explanations with Information Bottleneck. In *International Conference on Machine Learning*, 32062–32082. PMLR.

Liu, Z.; ZHANG, Y.; Wang, T.; Wang, Z.; Luo, D.; Du, M.; Wu, M.; Wang, Y.; Chen, C.; Fan, L.; and Wen, Q. 2024c. Explaining Time Series via Contrastive and Locally Sparse Perturbations. In *International Conference on Learning Representations*.

Louizos, C.; Welling, M.; and Kingma, D. P. 2018. Learning Sparse Neural Networks through L_0 Regularization. In *International Conference on Learning Representations*.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. arXiv:1611.00712.

Miao, S.; Liu, M.; and Li, P. 2022. Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism. In *Proceedings of the 39th International Conference on Machine Learning*, 15524–15543. PMLR.

Ming, Y.; Xu, P.; Qu, H.; and Ren, L. 2019. Interpretable and Steerable Sequence Learning via Prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 903–913. Association for Computing Machinery.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*.

Peng, Z.; Ren, S.; Gu, X.; Yang, L.; Wang, X.; and Sun, L. 2026. ProtoTS: Learning Hierarchical Prototypes for Explainable Time Series Forecasting. In *International Conference on Learning Representations*.

Queen, O.; Hartvigsen, T.; Koker, T.; He, H.; Tsiligkaridis, T.; and Zitnik, M. 2023. Encoding Time-Series Explanations through Self-Supervised Model Behavior Consistency. In *Advances in Neural Information Processing Systems*, volume 36, 32129–32159. Curran Associates, Inc.

Rasp, S.; Dueben, P. D.; Scher, S.; Weyn, J. A.; Mouatadid, S.; and Thuerey, N. 2020. WeatherBench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11): e2020MS002203.

Rudin, C. 2019a. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5): 206–215.

Rudin, C. 2019b. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5): 206–215.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*, 3319–3328. PMLR.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33: 6827–6839.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. arXiv:physics/0004057.

Tomioka, H.; and Yoshimura, G. 2026. PatchDecomp: Interpretable Patch-Based Time Series Forecasting. arXiv:2603.03902.

van Sprang, A.; Acar, E.; and Zuidema, W. 2025. Interpretability for Time Series Transformers using A Concept Bottleneck Framework. arXiv:2410.06070.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zheng, X.; Shirani, F.; Chen, Z.; Lin, C.; Cheng, W.; Guo, W.; and Luo, D. 2025. F-Fidelity: A Robust Framework for Faithfulness Evaluation of Explainable AI. In *The Thirteenth International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Theoretical Analysis: Full Proofs

Assumptions

This appendix proves the results of Section Theoretical Analysis. We first record, as (A1)–(A3), the standing assumptions stated in prose in the Section Preliminaries and at the Section Theoretical Analysis. All results below are understood under these assumptions. Throughout, $n = HC$ denotes the dimension of the future Y, and all differential entropies are assumed finite.

- **(A1)** *Determined context.* $\mathcal{C} = c(X) = (\tau;\, \mu, \sigma;\, Q)$ is a measurable function of the input window and its timestamp.
- **(A2)** *Invertible decomposition.* Once (τ, μ, σ, Q) are fixed, the decomposition of Eq. (2) is invertible, so given $\mathcal{C}$ the window X and its remainder $\Delta = D$ determine one another. The structural frame extends periodically over the forecast horizon (Eq. (8)), so given $\mathcal{C}$ the future Y and its remainder Δ_Y likewise determine one another.
- **(A3)** *Exogenous gate noise.* The explanation is generated from the remainder and the context together with independent noise, $E = e(\Delta, \mathcal{C}, U)$, where U collects the logistic noises of the sampler in Eq. (6) and is independent of (X, Y).

Lemma

To provide the analysis, we first introduce an auxiliary lemma for the proofs.

Lemma 2 (Conditional Markov structure)**.** *Suppose that assumptions* (A1)–(A3) *hold, we have the following three properties:*

- *The chain $Y \leftrightarrow \Delta \leftrightarrow E$ holds given $\mathcal{C}$, hence $I(E; Y \mid \mathcal{C}) \le I(\Delta; Y \mid \mathcal{C})$.*
- *For fixed selective explanations $E_{\mathcal{M}} = (\mathcal{M}, \Delta_{\mathcal{M}})$ with $\mathcal{M} \subseteq \mathcal{M}'$, $I(E_{\mathcal{M}}; Y \mid \mathcal{C}) \le I(E_{\mathcal{M}'}; Y \mid \mathcal{C})$.*
- *Relevance to the future equals relevance to its deseasonalized remainder, $I(E; Y \mid \mathcal{C}) = I(E; \Delta_Y \mid \mathcal{C})$.*

Proof. By (A3), E is generated from $(\Delta, \mathcal{C})$ and exogenous noise independent of Y. Hence $Y \perp E \mid (\Delta, \mathcal{C})$, which is the conditional Markov chain. Applying the chain rule to $I((E, \Delta); Y \mid \mathcal{C})$ in both orders yields

$$I(E; Y \mid \mathcal{C}) + I(\Delta; Y \mid E, \mathcal{C}) = I(\Delta; Y \mid \mathcal{C}) + I(E; Y \mid \Delta, \mathcal{C}).$$

The last term is zero by conditional independence and the second term is nonnegative, so $I(E; Y \mid \mathcal{C}) \le I(\Delta; Y \mid \mathcal{C})$. If $\mathcal{M} \subseteq \mathcal{M}'$, then $E_{\mathcal{M}}$ is a deterministic projection of $E_{\mathcal{M}'}$. The same data-processing argument gives $I(E_{\mathcal{M}}; Y \mid \mathcal{C}) \le I(E_{\mathcal{M}'}; Y \mid \mathcal{C})$.

For the last identity, (A2) extends the determined structure over the horizon, so for each fixed context value c the subtraction $Y \mapsto \Delta_Y$ is a bijection with measurable inverse. Data processing applied in both directions shows that an invertible transformation of one argument leaves mutual information unchanged, so $I(E; Y \mid \mathcal{C} = c) = I(E; \Delta_Y \mid \mathcal{C} = c)$ for every c; averaging over $\mathcal{C}$ gives $I(E; Y \mid \mathcal{C}) = I(E; \Delta_Y \mid \mathcal{C})$. □

We first restate the exact rate-saving result, which motivates conditioning the information bottleneck on the disclosed context.

Theorem 3 (Exact rate saving from disclosing the context)**.** *For any explanation E, $I(X; E) = I(\mathcal{C}; E) + I(\Delta; E \mid \mathcal{C})$, so the rate saved by disclosing the context is exact:*

$$\Delta R(E) \;:=\; I(X; E) - I(\Delta; E \mid \mathcal{C}) \;=\; I(\mathcal{C}; E) \;\ge\; 0,$$

with equality if and only if E is independent of $\mathcal{C}$. Moreover, the saving is at least the predictive information routed through the context, $I(\mathcal{C}; E) \;\ge\; I(E; Y) - I(E; Y \mid \mathcal{C})$.

Conditioning on $\mathcal{C}$ prevents double-counting: subtracting $I(\mathcal{C}; E)$ from the rate means the explanation is not penalized for information already present in the context, while the encoder remains free to adapt to $\mathcal{C}$. The final inequality exposes the flaw of unconditional explanations, which to obtain predictive accuracy ($I(E; Y) \ge I(X; Y) - \varepsilon$) must waste a portion of the rate ($I(\mathcal{C}; Y) - \varepsilon$) reproducing the shared structure.

Proof of Theorem 3

Proof. By (A1), $\mathcal{C} = c(X)$ is a measurable function of X, so the context is degenerate conditionally on X and mutual information with it vanishes:

$$I((X, \mathcal{C}); E) = I(X; E) + I(\mathcal{C}; E \mid X) = I(X; E).$$

Expanding the same quantity with $\mathcal{C}$ first,

$$I((X, \mathcal{C}); E) = I(\mathcal{C}; E) + I(X; E \mid \mathcal{C}).$$

By (A2), for each fixed context value c the determined structure is a fixed quantity, so the map $\varphi_c : X \mapsto \Delta$ that subtracts it is a measurable bijection with measurable inverse $\varphi_c^{-1} : \Delta \mapsto X$. Deterministic transformations of one argument cannot increase mutual information, applied in both directions: $\Delta = \varphi_c(X)$ gives $I(\Delta; E \mid \mathcal{C} = c) \le I(X; E \mid \mathcal{C} = c)$, and $X = \varphi_c^{-1}(\Delta)$ gives the reverse inequality, hence equality for every c. Averaging over $\mathcal{C}$,

$$I(X; E \mid \mathcal{C}) = I(\Delta; E \mid \mathcal{C}).$$

Combining the previous steps proves the decomposition, and therefore

$$\Delta R(E) = I(X; E) - I(\Delta; E \mid \mathcal{C}) = I(\mathcal{C}; E) \ge 0,$$

with equality if and only if $E \perp \mathcal{C}$, since mutual information vanishes exactly under independence.

Furthermore, expand $I(E; (\mathcal{C}, Y))$ by the chain rule in both orders:

$$\begin{aligned} I(E; \mathcal{C}) + I(E; Y \mid \mathcal{C}) &= I(E; (\mathcal{C}, Y)) \\ &= I(E; Y) + I(E; \mathcal{C} \mid Y). \end{aligned}$$

Rearranging and using the nonnegativity of conditional mutual information,

$$\begin{aligned} I(E; \mathcal{C}) &= I(E; Y) - I(E; Y \mid \mathcal{C}) + I(E; \mathcal{C} \mid Y) \\ &\ge I(E; Y) - I(E; Y \mid \mathcal{C}), \end{aligned}$$

with equality if and only if $E \perp \mathcal{C} \mid Y$.

Repeating Steps 1–3 with Y in place of E gives $I(X;Y) = I(\mathcal{C};Y) + I(\Delta;Y \mid \mathcal{C})$. If E is ε-sufficient, $I(E;Y) \geq I(X;Y) - \varepsilon$, then combining Step 4 with the ceiling $I(E;Y \mid \mathcal{C}) \leq I(\Delta;Y \mid \mathcal{C})$ of Lemma 2,

$$\begin{aligned} I(\mathcal{C};E) &\geq I(E;Y) - I(E;Y \mid \mathcal{C}) \\ &\geq \big(I(X;Y) - \varepsilon\big) - I(\Delta;Y \mid \mathcal{C}) \\ &= I(\mathcal{C};Y) - \varepsilon. \end{aligned}$$

This is the claim in the discussion following Theorem 3: to be ε-sufficient, an unconditional explanation must spend at least $I(\mathcal{C};Y) - \varepsilon$ of its rate reproducing structure already disclosed by the context. □

Proof of Theorem 1

Proof. Part 1 (the error floor). For fixed (e, c), the conditional mean $\bar{y}_{e,c} = \mathbb{E}[Y \mid E = e, \mathcal{C} = c]$ minimizes squared error, so every predictor $g(e;c)$ satisfies

$$\mathbb{E}[\|Y - g(e;c)\|_2^2 \mid e, c] \geq \operatorname{tr} \Sigma_{e,c},$$

where $\Sigma_{e,c}$ is the conditional covariance of Y. The Gaussian maximizes entropy among distributions with covariance $\Sigma_{e,c}$, and the arithmetic–geometric mean inequality gives

$$h(Y \mid e, c) \leq \frac{n}{2} \log\left(2\pi e \, \frac{\operatorname{tr} \Sigma_{e,c}}{n}\right).$$

Equivalently,

$$\operatorname{tr} \Sigma_{e,c} \geq \frac{n}{2\pi e} \exp\left(\frac{2}{n} h(Y \mid e, c)\right).$$

Taking expectations over $(E, \mathcal{C})$ and applying Jensen's inequality to the convex exponential,

$$\mathbb{E}\|Y - \hat{Y}\|_2^2 \geq \frac{n}{2\pi e} \exp\left(\frac{2}{n} h(Y \mid E, \mathcal{C})\right).$$

Finally, $h(Y \mid E, \mathcal{C}) = h(Y \mid \mathcal{C}) - I(E;Y \mid \mathcal{C})$, which gives the stated bound.

Part 2 (monotonicity in the budget). Because $E_{\mathcal{M}}$ is measurable with respect to $E_{\mathcal{M}'}$, every predictor using $(E_{\mathcal{M}}, \mathcal{C})$ is also a valid predictor using $(E_{\mathcal{M}'}, \mathcal{C})$. The conditional expectation $\mathbb{E}[Y \mid E_{\mathcal{M}'}, \mathcal{C}]$ is the L^2 projection of Y onto the larger predictor space, and projection error cannot increase when the space is enlarged. This gives the displayed inequality; since any larger budget admits every selection feasible under a smaller one, the minimal Bayes risk is nonincreasing in the budget. □

Dataset statistics

Table 4 summarizes the used benchmarks, all standard public multivariate long-term-forecasting datasets:

- *ETTh1, ETTh2, ETTm1, ETTm2* (Zhou et al. 2021) record two years (July 2016 to July 2018) of Electricity Transformer Temperature data from two transformers in two Chinese counties. Each has 7 channels, the target oil temperature plus six external power-load features; the `h` variants are sampled hourly and the `m` variants every 15 minutes.
- *Weather* contains 21 meteorological indicators, such as air temperature, humidity, and wind speed, recorded every 10 minutes throughout 2020 at the Max Planck Institute for Biogeochemistry weather station in Jena, Germany.

All splits and preprocessing follow the TQNet protocol verbatim: the ETT family is split 6:2:2 and the remaining datasets 7:1:2 chronologically; every channel is z-score standardized with statistics computed on the training split only, and all metrics are computed on the standardized data. Windows are formed by sliding over each split with look-back L=96 and horizons $H \in \{96, 192, 336, 720\}$, and each window's timestamp provides the phase index τ of Eq. (2). The cycle length W of the seasonal profile Q is fixed from the sampling rate, following the CycleNet convention (Lin et al. 2024): one day for the ETT family, and Weather (W=24, 96, 144, and 144 respectively).

Baseline details

This section describes each baseline. The information contains what the method is, what it emits as an explanation where one exists, and how it is configured or adapted for our protocol.

Black-box forecasters. The black-box group establishes the accuracy state of the art; these models offer no native explanation, so they appear in the accuracy comparison only, with their numbers taken from the TQNet consolidated table under the identical fixed-lookback protocol rather than rerun.

- *TQNet* (Lin et al. 2025) captures multivariate correlations with periodically shifted learnable temporal queries in a single-layer attention over the raw input, followed by a lightweight MLP; it is the protocol anchor whose training pipeline our accuracy evaluation reuses.
- *CycleNet* (Lin et al. 2024) models the recurring periodic pattern of each channel with learnable recurrent cycles and forecasts the residual with a linear or MLP backbone; our learned seasonal profile Q adopts this mechanism, as credited in §.
- *iTransformer* (Liu et al. 2024a) inverts the Transformer: each channel's whole look-back is embedded as one variate token, attention runs across variate tokens to model channel dependence, and the feed-forward network models the temporal dimension.
- *PatchTST* (Nie et al. 2023) is a channel-independent Transformer over patch tokens, with each channel processed as a separate sequence of sub-series patches under shared weights.
- *TimesNet* (Wu et al. 2023) folds the 1D series into a set of 2D tensors along its dominant periods and models intra- and inter-period variation with inception-style 2D convolutions.

Self-interpretable forecasters. These models are trained on the same data as IB-Forecast and are the direct competitors. Each is scored on forecasting accuracy, and its native explanation, where one exists, is scored on the fidelity metrics against its own forecasts.

Table 4: Statistics of the multivariate benchmarks.

Dataset	Channels	Time steps	Frequency	Split
ETTh1, ETTh2	7	17,420	1 hour	6:2:2
ETTm1, ETTm2	7	69,680	15 min	6:2:2
Weather	21	52,696	10 min	7:1:2
Electricity	321	26,304	1 hour	7:1:2
Solar	137	52,560	10 min	7:1:2
Traffic	862	17,544	1 hour	7:1:2

- *TFT* (Lim et al. 2021) forecasts with variable-selection networks and interpretable attention. We implement its interpretable core faithfully: a per-timestep variable-selection network of gated residual networks that emits softmax weights over channels, an LSTM encoder, and an interpretable multi-head attention in which all heads share a single value projection so that the head-averaged attention matrix is directly readable. Components absent from our benchmarks, namely static covariates, known-future inputs, and the quantile head, are omitted; the model is trained with MSE like every other model in the harness, and instance normalization is added, without which TFT-era models underfit the distribution shift of the long-horizon benchmarks. Its native importance map is the product of the temporal attention, averaged over heads and horizon steps, and the per-timestep variable-selection weight, exactly the two quantities its paper reads off as interpretability outputs.
- *PatchDecomp* (Tomioka and Yoshimura 2026) writes the forecast as an exact sum of per-patch contributions. Our reimplementation matches the official architecture: a per-channel convolutional patch embedding with learned positional context, residual MLP encoders, a per-patch scalar bias, a single bias-free multi-head attention with future-patch queries over input-patch keys and values, and a bias-free linear decoder. Because every map after the attention is linear and bias-free, the forecast decomposes exactly into additive per-input-patch contributions, which are its native explanation; the importance of a (patch, channel) token is its mean absolute contribution over the horizon, broadcast to point resolution. Two differences from the official code are disclosed: our benchmarks carry no static or known-future covariates, so those encoders are dropped, and the real-data mode is channel-independent with shared weights, which is how its own LTSF experiments treat multivariate data.
- *DLinear* (Zeng et al. 2023) decomposes the input into a moving-average trend and a remainder and forecasts each with a per-channel linear map. The model is fully transparent, and its native attribution is exact in closed form, since each future value is a weighted sum of past values with known weights.
- *ProtoTS* (Peng et al. 2026) explains by similarity to hierarchical prototypes, which has no input-attribution analogue, so it joins the accuracy comparison only. It is run from its official release, channel-independently under the anchor protocol, which is the one adaptation its multivariate application requires, keeping its two-stage training and prototype-splitting schedule with release hyperparameters and best-validation selection throughout. Its per-cell training cost grows quickly with channels and horizon: the Weather cells exceeded a 10-hour single-GPU budget mid-training and are excluded from Table 1 rather than reported from partially trained models, and the ETTm2 $H{=}720$ cell hit the same cap at epoch 56 of 60 with its learning rate at the floor and validation flat for several epochs, so it is reported from its best-validation checkpoint, the same selection the completed run would have used.

TFT and PatchDecomp receive per-dataset validation tuning identical to our own model's, and each native explanation is scored on its own model where fidelity is concerned.

Gradient-based post-hoc explainers. All three explain our trained model. Because the deployed model thresholds its gates at inference and gradients do not flow through hard gates, the gradient explainers are computed on the model's soft relaxation, with gates equal to their sigmoid probabilities, which is the standard smooth-surrogate assumption these methods make. The attribution target is the scalar forecast energy $\frac{1}{2}\|\hat{Y}\|_2^2$.

- *Saliency* (Simonyan, Vedaldi, and Zisserman 2014) is the absolute input gradient of the target.
- *Integrated Gradients* (IG; Sundararajan, Taly, and Yan 2017) integrates gradients along the straight path from a baseline to the input over 32 steps; the baseline is the per-channel window mean, an in-distribution reference, and the attribution is the path-averaged gradient times the input-minus-baseline displacement.
- *TIMING* (Jang, Kim, and Yang 2025) is temporality-aware IG: at each interpolation step, random contiguous time segments, one random channel each, are held at the true input and excluded from both the interpolation and the gradient, so every gradient is taken at a point that is on-manifold in the fixed regions, and the path sum is normalized per cell by how often that cell was free. Our port follows the reference implementation and uses the same target and baseline as the IG row, so the difference between the two rows isolates the estimator. Its release defaults, 50 segments of minimum length 10 tuned for MIMIC-III's 48-step windows, saturate a 96-step window; the TIMING row of Table 2 therefore reports the best of the defaults and two segment-count/length variants per dataset, which is favorable treatment, and the defaults alone score 0.404/0.450/0.771 on ETTh1/ETTm1/Weather.

Perturbation-based post-hoc explainers. *Occlusion* is gradient-free and runs on the real hard-gated model: each (patch, channel) block is replaced by the per-channel reference, the squared forecast shift is recorded as that block's importance, and the score is broadcast back to point resolution; the block length equals the model's own patch length. *Random* assigns uniform-random importance and serves as the floor of the fidelity metrics.

Table 5: Adopted IB-Forecast configuration on all datasets. All cells share a fixed backbone: hard straight-through gates (temperature 0.5, annealed), gate width d=64, patch length 12, RevIN on, per-gate prior π=0.5, and 30 Adam epochs.

Dataset	Readout	Gate layers	Learning rate	β	Cycle W	Cycle used
ETTh1	gated-linear	1	1×10^{-2}	0.02	24	yes
ETTh2	gated-linear	1	1×10^{-2}	0.02	24	yes
ETTm1	gated-linear	2	1×10^{-2}	0.02	96	yes
ETTm2	gated-linear	2	1×10^{-2}	0.02	96	yes
Weather	MLP	1	3×10^{-3}	0.02	144	yes
Electricity	MLP	1	3×10^{-3}	0.02	168	yes
Solar	MLP	1	3×10^{-3}	0.1	144	yes
Traffic	MLP	1	3×10^{-3}	0.02	168	yes

Test-time mask optimizers. Dynamask (Crabbé and Van Der Schaar 2021) and ExtremalMask (Enguehard 2023) learn a per-instance mask by optimizing directly against the model's own output, so they are near-oracles on fidelity metrics by construction. Both are published for classification; we re-target them to forecasting by replacing the class-probability-preservation term of their objectives with forecast preservation, the MSE over the full $H \times C$ forecast, with gradients through the same soft relaxation as above.

- *Dynamask* optimizes per-instance mask logits with Adam under its published extremal keep-ratio constraint, the sorted-mask regularizer, which lets us pin the mask to the same budget as the native mask, together with its temporal total-variation term; the budget pressure is annealed from soft to binding over the run. Its published perturbation operator fades masked values to a per-channel moving average; when scoring, we instead hand it the evaluation protocol's own replacement reference, so it optimizes against exactly the operator the metric perturbs with, the most favorable setting.
- *ExtremalMask* keeps the same budget machinery but learns the replacement values jointly with the mask: a small convolutional network proposes replacements around the neutral anchor and is ℓ_2-regularized toward it, so the mask cannot hide information in the perturbation.

Both run 500 optimization steps per test batch with the same or an oracle sparsity budget, and § verifies their scores are converged rather than under-optimized.

Implementation

The gate network is 1 to 2 Transformer layers with d=64 and patch length 12 on real data, using channel-independent gate attention for large C. The base readout is a gated linear map or MLP chosen on validation. Budget sparsity is ρ=0.2 for explanation studies, and β=0.02 KL elsewhere. Optimization uses Adam, with the gate temperature T of Eq. (6) annealed over training. Headline results use the protocol seed 2024, and for robustness every headline cell is additionally run with 3 seeds, with a seed standard deviation of 0.005 MSE or less on 17 of 20 cells and a maximum of 0.010. All real-data training fits on a single RTX 4090 GPU. We further provide all the hyperparameter settings in Table 5.

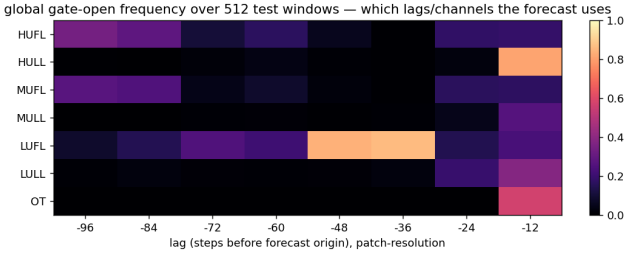

Figure 4: Gate-open frequency per (lag-patch, channel) over 512 ETTh1 test windows (ρ=0.2, H=96).

Factor decomposition

For controlled studies, the gate head emits K+1 maps. Map 0 is the base mask of the main text, and maps 1 through K feed a masked-replacement pooling into factor scores s_k and per-channel horizon loadings, $\hat{D}^{\text{struct}}_{h,c} = \sum_k A_{h,k,c}\, s_k$, giving a horizon-resolved account of which support drives which part of the future. On the synthetic benchmarks with planted multi-driver structure, this decomposition is recoverable. On the real benchmarks, the factor path collapses and contributes exactly zero, since zeroing A changes neither accuracy nor any faithfulness score (Table 3), so the real-data model descopes it and the deployed explanation.

Global gate-usage statistics

Figure 4 aggregates the mask over 512 ETTh1 test windows, giving the gate-open frequency per (lag-patch, channel). The daily structure, at lag -96 to -84 and the most recent patches, and the exclusion of near-constant channels emerge without supervision, which is the population-level counterpart of the single-window case study in Figure 3.

Per-horizon accuracy results

Table 6 expands Table 1 of the main text to full per-horizon MSE and MAE on the five anchor datasets, with the same interpretable/black-box grouping. IB-Forecast cells are means over 3 seeds, from 2024 to 2026, and the seed standard deviation is not greater than 0.005 MSE on 17 of 20 cells, with a maximum of 0.010 at ETTh2, H=720. PatchDecomp, TFT, and ProtoTS cells are our runs at the protocol seed

Table 6: Per-horizon forecasting results under the TQNet protocol (look-back 96, standardized metrics), extending Table 1: each Avg row is the mean of the four horizons above it and reproduces the corresponding row of Table 1. Bold marks the best overall per column pair, and underline marks the best interpretable model. [§]ProtoTS Weather cells exceeded the 10-hour-per-cell compute budget and are omitted, so its Average row is also omitted.

		Interpretable-by-design										*Black-box*									
		IB-Forecast		PatchDecomp		TFT		DLinear		ProtoTS[§]		TQNet		CycleNet		iTransformer		PatchTST		TimesNet	
Dataset	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.382	**0.393**	0.373	0.397	0.506	0.482	0.386	0.400	0.414	0.419	**0.371**	**0.393**	0.375	0.395	0.386	0.405	0.414	0.419	0.384	0.402
	192	0.426	**0.418**	**0.424**	0.423	0.569	0.507	0.437	0.432	0.555	0.490	0.428	0.426	0.436	0.428	0.441	0.436	0.460	0.445	0.436	0.429
	336	**0.466**	**0.438**	0.494	0.464	0.613	0.523	0.481	0.459	0.722	0.579	0.476	0.446	0.496	0.455	0.487	0.458	0.501	0.466	0.491	0.469
	720	**0.477**	**0.468**	0.532	0.501	0.636	0.545	0.519	0.516	0.713	0.596	0.487	0.470	0.520	0.484	0.503	0.491	0.500	0.488	0.521	0.500
	Avg	**0.438**	**0.429**	0.456	0.446	0.581	0.514	0.456	0.452	0.601	0.521	0.441	0.434	0.457	0.441	0.454	0.448	0.469	0.455	0.458	0.450
ETTh2	96	**0.291**	**0.340**	0.295	0.347	0.386	0.402	0.333	0.387	0.306	0.347	0.295	0.343	0.298	0.344	0.297	0.349	0.302	0.348	0.340	0.374
	192	0.372	**0.391**	0.397	0.413	0.482	0.454	0.477	0.476	0.397	0.408	**0.367**	0.393	0.372	0.396	0.380	0.400	0.388	0.400	0.402	0.414
	336	0.431	0.432	0.439	0.448	0.469	0.459	0.594	0.541	0.455	0.454	**0.417**	**0.427**	0.431	0.439	0.428	0.432	0.426	0.433	0.452	0.452
	720	0.430	**0.443**	0.463	0.473	0.454	0.462	0.831	0.657	0.451	0.461	0.433	0.446	0.450	0.458	**0.427**	0.445	0.431	0.446	0.462	0.468
	Avg	0.381	**0.402**	0.399	0.420	0.448	0.444	0.559	0.515	0.402	0.418	**0.378**	**0.402**	0.388	0.409	0.383	0.407	0.387	0.407	0.414	0.427
ETTm1	96	0.326	0.364	0.322	0.361	0.421	0.436	0.345	0.372	0.336	0.361	**0.311**	**0.353**	0.319	0.360	0.334	0.368	0.329	0.367	0.338	0.375
	192	0.364	0.388	0.369	0.388	0.480	0.465	0.380	0.389	0.383	0.387	**0.356**	**0.378**	0.360	0.381	0.377	0.391	0.367	0.385	0.374	0.387
	336	0.395	0.411	0.390	0.407	0.482	0.464	0.413	0.413	0.724	0.562	0.390	**0.401**	**0.389**	0.403	0.426	0.420	0.399	0.410	0.410	0.411
	720	0.448	0.447	0.454	0.440	0.566	0.512	0.474	0.453	0.748	0.580	0.452	0.440	**0.447**	0.441	0.491	0.459	0.454	**0.439**	0.478	0.450
	Avg	0.383	0.403	0.384	0.399	0.487	0.469	0.403	0.407	0.548	0.473	**0.377**	**0.393**	0.379	0.396	0.407	0.410	0.387	0.400	0.400	0.406
ETTm2	96	0.167	0.248	0.179	0.266	0.197	0.276	0.193	0.292	0.181	0.256	0.173	0.256	**0.163**	**0.246**	0.180	0.264	0.175	0.259	0.187	0.267
	192	0.232	0.291	0.250	0.314	0.273	0.325	0.284	0.362	0.254	0.305	0.238	0.298	**0.229**	**0.290**	0.250	0.309	0.241	0.302	0.249	0.309
	336	0.291	0.331	0.331	0.373	0.326	0.356	0.369	0.427	0.321	0.349	0.301	0.340	**0.284**	**0.327**	0.311	0.348	0.305	0.343	0.321	0.351
	720	0.398	0.394	0.427	0.423	0.447	0.424	0.554	0.522	0.854	0.624	0.397	0.396	**0.389**	**0.391**	0.412	0.407	0.402	0.400	0.408	0.403
	Avg	0.272	0.316	0.297	0.344	0.311	0.345	0.350	0.401	0.403	0.384	0.277	0.323	**0.266**	**0.314**	0.288	0.332	0.281	0.326	0.291	0.333
Weather	96	0.159	0.204	0.170	0.212	0.181	0.230	0.196	0.255	—	—	**0.157**	**0.200**	0.158	0.203	0.174	0.214	0.177	0.210	0.172	0.220
	192	0.208	0.247	0.217	0.253	0.242	0.280	0.237	0.296	—	—	**0.206**	**0.245**	0.207	0.247	0.221	0.254	0.225	0.250	0.219	0.261
	336	0.264	0.289	0.273	0.293	0.293	0.315	0.283	0.335	—	—	**0.262**	**0.287**	**0.262**	0.289	0.278	0.296	0.278	0.290	0.280	0.306
	720	**0.344**	0.344	0.351	0.345	0.382	0.367	0.345	0.381	—	—	**0.344**	0.342	**0.344**	0.344	0.358	0.349	0.354	**0.340**	0.365	0.359
	Avg	0.244	0.271	0.253	0.276	0.275	0.298	0.265	0.317	—	—	**0.242**	**0.269**	0.243	0.271	0.258	0.278	0.259	0.273	0.259	0.287
Overall		0.344	**0.364**	0.358	0.377	0.420	0.414	0.407	0.418	—	—	**0.343**	**0.364**	0.346	0.366	0.358	0.375	0.356	0.372	0.364	0.380

and the DLinear and black-box columns are from the TQNet paper's consolidated table.

Breadth on large-channel datasets

We also evaluate IB-Forecast on three large channel datasets: Electricity, Solar, and Traffic. These experiments follow the same protocol as the main results. Table 7 reports results over 3 seeds and 4 horizons, with seed standard deviation no larger than 0.008, against TQNet's published cells. On Electricity and Solar, IB-Forecast is essentially at parity: the average gap is $+2.4\%$ and $+3.0\%$, the per-horizon gap ranges from -0.6% to $+6.0\%$ and from -1.9% to $+7.5\%$, and IB-Forecast wins outright at Electricity $H{=}192$ and Solar $H{=}336$. The one substantive gap is Traffic ($C{=}862$), at $+8.1$ to $+11.4\%$ per horizon and $+9.0\%$ on average, shrinking from $+11.4\%$ at $H{=}96$ to about $+8\%$ at longer horizons. A capacity study shows that this gap is not caused by the gate network size, since doubling its width or depth only changes results in the third decimal place. These results provide breadth and scalability evidence. In particular, Traffic trains in only 2.6GB with channel-independent gate attention, showing that the explanation machinery can scale to hundreds of channels.

Hyperparameter sensitivity

The deployed model has five regularization hyperparameters: the information-bottleneck weight β, the gate temporal-variation weight λ_{tv}, the budget-penalty weight λ_b, the open-rate target ρ, and the per-gate prior π. Figure 5 sweeps each one at a time about the shared default ($\beta{=}0.02$, $\lambda_{tv}{=}10^{-3}$, $\lambda_b{=}5$, $\rho{=}0.2$, $\pi{=}0.2$) on ETTh1, ETTm1 and Weather at $H{=}96$, reporting the change in test MSE relative to the default, the realized base-mask open rate, and the matched-budget fidelity score.

Two patterns stand out. First, **accuracy is essentially insensitive to the regularizers**: across every sweep of β, λ_{tv}, λ_b and π the MSE stays within about 3% of the default on all three datasets (top row), so none of these weights needs per-dataset tuning. The only hyperparameter that moves accuracy materially is ρ, and it does so because it *is* the budget—the Pareto dial of Section , not a nuisance parameter. Second, **the regularizers act on the explanation rather than the forecast**: ρ sets the open rate almost linearly (middle row), while β, λ_b and π shift it more gently, and the fidelity score (bottom row) stays high, typically above 0.9. It falls only at the extremes, where a too-large β or π over-closes the mask on the ETT datasets or a too-small ρ collapses it fully closed (the

Table 7: Experiments on large-channel datasets: IB-Forecast vs. TQNet. IB-Forecast MSE is the mean value of 3 seeds with std ≤ 0.008. TQNet is the published value. Δ is the relative gap. A positive value means IB-Forecast has the higher (worse) MSE.

	Electricity (C=321)			Solar (C=137)			Traffic (C=862)		
H	IB-Forecast	TQNet	Δ	IB-Forecast	TQNet	Δ	IB-Forecast	TQNet	Δ
96	0.136	0.134	+1.5%	0.186	0.173	+7.5%	0.460	0.413	+11.4%
192	0.153	0.154	−0.6%	0.200	0.199	+0.5%	0.467	0.432	+8.1%
336	0.171	0.169	+1.2%	0.207	0.211	−1.9%	0.487	0.450	+8.2%
720	0.213	0.201	+6.0%	0.221	0.209	+5.7%	0.527	0.486	+8.4%
Avg	0.168	0.164	+2.4%	0.204	0.198	+3.0%	0.485	0.445	+9.0%

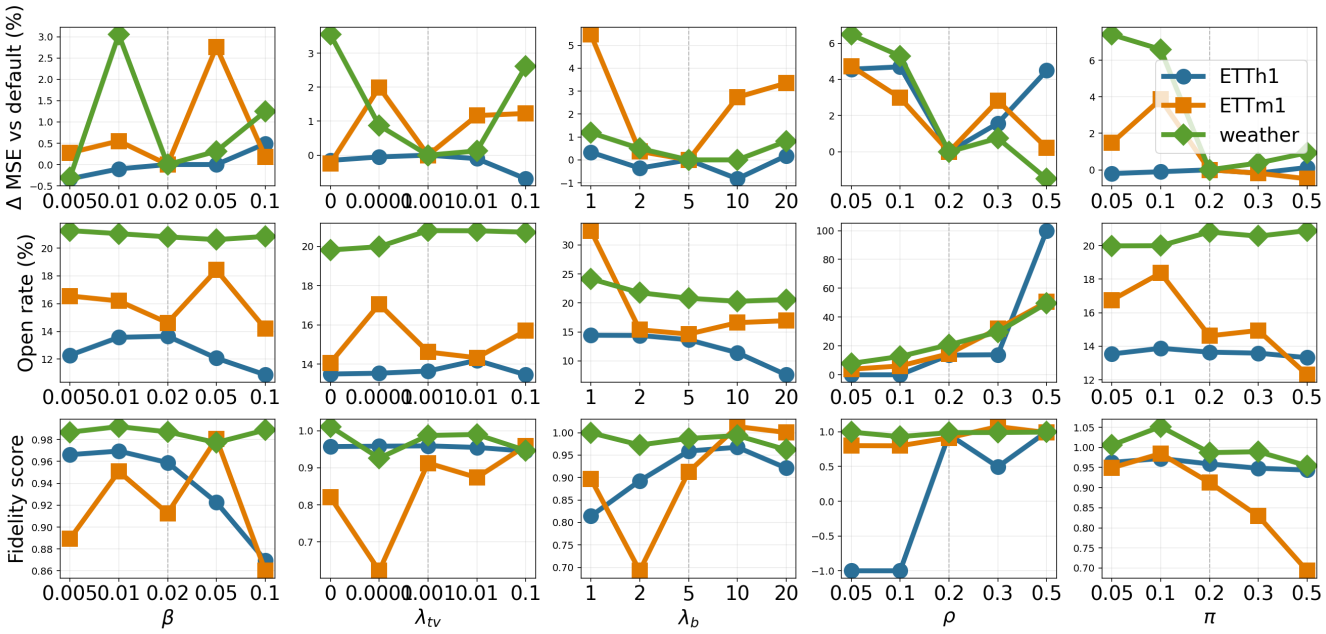

Figure 5: One-at-a-time hyperparameter sensitivity on ETTh1, ETTm1 and Weather (H=96, seed 2024). Columns are the five regularization hyperparameters; the dashed line marks the shared default. Rows: change in test MSE relative to the default (%), realized base-mask open rate (%), and matched-budget fidelity score. Accuracy (top) is flat except under ρ, the budget dial; the hyperparameters instead move the explanation's sparsity (middle) and faithfulness (bottom), both stable across a wide interior.

$\rho \leq 0.1$ ETTh1 failure noted in Section), plus a few isolated dips on ETTm1, the noisiest of the three. The default therefore sits in a broad interior region where accuracy, sparsity and faithfulness are simultaneously stable, which is why a single configuration transfers across datasets in Tables 1 and 2.

Synthetic ground-truth support recovery

In this section, we report the perturbation-free faithfulness evidence that corroborates the real-data result of the matched-budget evaluation (Table 2). Real-data fidelity necessarily perturbs inputs and therefore inherits protocol judgment calls; here the generator itself plants the causal support, so explanations are scored against known ground truth with no perturbation operator at all.

Benchmark construction. The benchmark instantiates the structural assumptions of Problem 1 directly, so that every disclosed component of the model has a legitimate, measurable job. Each sample is a look-back window $X \in \mathbb{R}^{L\times C}$ with a multivariate target $Y \in \mathbb{R}^{H\times C}$, here L=96, H=24, C=4. For each window we draw a per-channel level $\mu_c \sim U(-3,3)$ and scale $s_c \sim U(0.5,1.5)$, a shared cycle phase $\tau \sim U\{0,\dots,W-1\}$ with W=24, and, independently per channel, a driver indicator $z_c \sim \text{Bernoulli}(\frac{1}{2})$. The window and its future are generated as

$$X[t,c] = \mu_c + s_c\Big(\phi_c\big((t+\tau) \bmod W\big)\beta_c(t) + z_c\, r_c(t) + \varepsilon^X_{t,c}\Big), \tag{11}$$

$$Y[t,c] = \mu_c + s_c\Big(\phi_c\big((t+L+\tau) \bmod W\big) + z_c\, g_c(t) + \varepsilon^Y_{t,c}\Big), \tag{12}$$

$$M[t,c] = z_c \cdot \mathbb{1}[t \in \text{supp}(r_c)]\,, \tag{13}$$

Table 8: Ground-truth support recovery on the framed benchmark (AUROC, AUR, and AUP). All cells are mean ± seed standard deviation over 3 seeds. Dynamask and ExtremalMask receive an oracle sparsity budget. The best performance is marked in bold. †TIMING runs its release defaults, whose segments saturate a 96-step window and score near chance.

Mode	**IB-Forecast**	PatchDecomp	TFT	Occlusion	Saliency	IG	TIMING†	Dynamask	ExtremalMask	Random
					AUROC					
`pulse`	**0.912±.003**	0.900±.002	0.549±.059	0.891±.002	0.846±.006	0.839±.003	0.524±.000	0.814±.001	0.868±.008	0.492±.000
`decoy`	**0.878±.005**	0.815±.008	0.493±.027	0.827±.005	0.769±.008	0.775±.008	0.530±.000	0.760±.015	0.796±.003	0.502±.000
`trend`	**0.689±.001**	0.554±.007	0.515±.011	0.650±.004	0.602±.007	0.637±.005	0.424±.000	0.559±.004	0.566±.007	0.501±.000
					AUR					
`pulse`	**0.791±.003**	0.618±.007	0.180±.035	0.568±.002	0.272±.022	0.230±.011	0.022±.000	0.366±.007	0.343±.005	0.492±.000
`decoy`	**0.755±.029**	0.565±.006	0.190±.068	0.416±.005	0.153±.037	0.152±.020	0.021±.001	0.267±.002	0.250±.008	0.502±.000
`trend`	**0.375±.015**	0.261±.007	0.150±.001	0.181±.005	0.068±.001	0.068±.001	0.012±.000	0.169±.001	0.159±.001	0.501±.000
					AUP					
`pulse`	0.471±.070	**0.755±.014**	0.193±.113	0.711±.004	0.409±.068	0.611±.062	0.061±.004	0.310±.010	0.294±.007	0.031±.000
`decoy`	0.193±.050	**0.638±.055**	0.068±.037	0.375±.009	0.127±.046	0.217±.058	0.065±.008	0.204±.003	0.194±.012	0.030±.000
`trend`	0.343±.003	0.201±.011	0.192±.007	**0.565±.008**	0.391±.007	0.520±.005	0.096±.006	0.156±.001	0.152±.002	0.133±.000

where every injected shape is de-meaned over the window, $\sum_t \phi_c(\cdot) = \sum_t \beta_c(t) = \sum_t r_c(t) = 0$, and $\varepsilon^X, \varepsilon^Y$ are i.i.d. Gaussian noise. Each term plays a role that mirrors one disclosed component of the model:

- *Frame* (μ_c, s_c): the per-channel level and scale, exactly the information the normalization frame (μ, σ of Eq. (2)) discloses. Because every injection is de-meaned, the window mean equals μ_c regardless of the driver, so the frame carries *no* driver information and frame and mask are orthogonal by construction.
- *Cycle* ϕ_c: a fixed per-channel periodic shape of period $W{=}24$, shared across the whole dataset, entering window and future at the same phase τ; this is the ground-truth counterpart of the learned profile Q. It mimics the periodicity of the real hourly benchmarks, where the look-back $L{=}96$ spans four days and the horizon $H{=}24$ spans one, and τ is handed to the model exactly as in the real protocol.
- *Background* β_c: a per-window mixture of random-period sinusoids, present in X but absent from Y (note β_c appears in Eq. (11) but not Eq. (12)), hence causally irrelevant by construction. A faithful explanation must leave it out.
- *Drivers* r_c (present iff $z_c{=}1$): the sparse in-channel causal event, and its type defines the mode. `pulse` plants a 6-step bump of amplitude $U(2,5)$ with random sign; `trend` plants a 24-step ramp with slope $U(0.05, 0.25)$ and random sign, whose realized rise the future response g_c carries and decays over the horizon; `decoy` plants the `pulse`. It then adds an attenuated, noise-corrupted copy elsewhere in the same channel. The copy is predictive of Y, but strictly inferior to the clean pulse. It is *excluded* from the ground truth M.

The forecast term g_c is the driver's effect on the future (for `trend`, the ramp's realized rise decaying over the horizon; for `pulse`/`decoy`, its analogous response). A channel with $z_c{=}0$ has a pure frame-and-cycle future, so a closed mask there asserts precisely that the channel followed the disclosed pattern, and by Eqs. (11)–(12) that assertion is true. The ground-truth mask M of Eq. (13) marks exactly the injected driver positions. Splits are 4000/1000/1000 windows; explanations are scored on 256 test windows with AUROC, AUP, and AUR against M.

Model and baselines. The model under explanation is the deployed architecture of the main text, with the frame, the cycle profile at $W{=}24$, and the base mask as the explanation, trained for 150 epochs at patch length 6, $\beta{=}0.1$, $\pi{=}0.05$ (three seeds; six where noted). The post-hoc baselines explain this same model. They are saliency, IG, TIMING, patch occlusion, and a random floor. Two are test-time mask optimizers, Dynamask and ExtremalMask. Both run at the oracle budget, namely the ground-truth mask's own density. TFT and PatchDecomp are native-versus-native comparisons: each is trained fresh on the benchmark in its strongest configuration (multivariate output with instance normalization), and its own explanation is read off exactly as its paper prescribes. Their architectures take no phase input, which we disclose; four full cycles are visible in every window, so the phase is inferable from the data they see.

Analysis. IB-Forecast ranks first in every AUROC cell in Table 8. It also performs especially well on recall. Its AUR is between 0.76 and 0.79 on `pulse` and `decoy`, while every baseline stays at or below 0.62. This shows that the budgeted mask recovers most of the planted support, whereas point scoring methods only identify fragments. The `decoy` setting gives a clearer causal test of the information bottleneck. PatchDecomp is close on the clean `pulse` setting, with 0.900 versus 0.912, but drops to 0.815 on `decoy` because its unconstrained forecaster uses the decoy. In contrast, our mask rejects the decoy and reaches 0.878. The effect is controlled by β. With $\beta{=}0.02$, `decoy` AUROC is only 0.09 to 0.36, which means the model actively selects the decoy. With $\beta{=}0.1$, it rises to 0.85 to 0.88. Spurious exclusion is therefore controlled by the objective, not by accident. The cycle component also matters. Removing it drops `pulse` AUROC from 0.876 to 0.631 and increases the mask's false positive mass from 0.68 to 0.955, because the budget is wasted on relearning periodic structure. TFT, by contrast, stays near chance in all modes, consistent with the attention is not explanation phenomenon

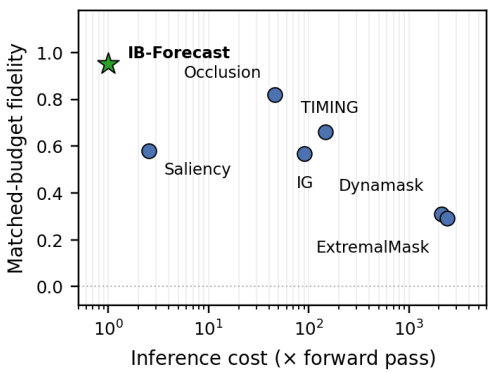

Figure 6: Fidelity against inference cost per explanation on the ETTh1 dataset at $H{=}96$, with cost on a logarithmic axis measured in forward-equivalent passes.

(Jain and Wallace 2019).

The results also show two costs. First, IB-Forecast trades precision for recall. It does not lead on AUP in Table 8, because the sufficiency objective tends to select extra tokens around each event instead of isolating only the exact causal tokens. The `trend` setting is difficult for every method. Our native explanation reaches 0.689, while the best post hoc method reaches 0.650. This happens because the target depends on the ramp's realized rise, so the truly sufficient evidence is smaller than the full annotation. As a result, all methods lose recall under this label. Second, the sufficiency constraint introduces an accuracy cost, and this benchmark makes that cost look large. PatchDecomp can read the whole window and reaches roughly half our MSE, for example 0.16 versus 0.30 on `pulse`. Under the real data protocol, however, the same constraint costs only about 3%. The synthetic setting is a worst case for a budgeted reader, since all useful signal is concentrated in the few tokens the model must find.

Efficiency analysis

We present the efficiency analysis in terms of the measured inference cost per explanation, timed on a GPU and expressed in forward-equivalent passes. As shown in Figure 6, among the post-hoc methods, saliency costs about 2.5, occlusion about 46, IG about 91, and TIMING about 148 forward-equivalent passes, and each trails the native mask by 0.13 to 0.37 fidelity. TFT and PatchDecomp also explain at near-zero cost but score negative fidelity. Figure 6 compares fidelity against inference cost per explanation. The native explanation is emitted by the forward pass and adds no computation. Among post-hoc methods, occlusion, the strongest on fidelity, requires 56 to 168 forward passes per window; TIMING requires 50 forward and backward passes and trails the native mask by 0.08 to 0.37 fidelity, with IG at 32 such passes and saliency at a single backward pass. The test-time mask optimizers spend 500 optimization steps per batch, each a forward and backward pass, and still trail by more than 0.5. TFT and PatchDecomp also explain at zero cost but score negative fidelity (Table 2), so low cost alone does not confer faithfulness. Finally, explanation scales to Traffic ($C{=}862$) with no memory overhead beyond the forecaster itself, at 2.6 GB total, via channel-independent gate attention.