CONTEX: REFORMULATING COUNTERFACTUAL INPUT GENERATION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep time series forecasting models achieve strong predictive performance, yet their inverse behavior remains opaque: they do not reveal the temporal drivers of desired alternative forecasts or how these inputs must change to reach them. Understanding this target-dependent inverse behavior enables auditing of learned temporal dependencies and, when appropriate feasibility constraints are available, can support decision-making. Existing counterfactual explanation methods address this through instance-wise optimization, which incurs high computational costs, inconsistent solutions across similar inputs, and limited real-time applicability. We address this as a target-conditioned inverse forecasting problem and introduce GenCF as a direct amortized realization of this formulation. Building on this perspective, we further improve transparency into a model's target-conditioned inverse behavior by separating model-level counterfactual explanations into target-relative temporal relevance and modification magnitude. To this end, we introduce ConTex, a forecaster-architecture-agnostic intervention function that maps an input and desired forecast trajectory to model-level counterfactual modifications. We further introduce target-conditioned intervention relevance, identifying which timesteps matter for inducing a desired forecast rather than retrospectively explaining the current prediction. ConTex operationalizes this notion through a learned temporal selection score and a corresponding modification strength, revealing where and how the input must change. Across multiple forecasting architectures and benchmark datasets, ConTex achieves highly competitive target validity while producing substantially sparser input counterfactuals. Additionally, ConTex reduces total computational cost by 12–36× and enables several-thousand-fold faster per-sample inference relative to instance-wise optimization. These results establish ConTex as a promising foundation for extending forecasting models with explanations of their inverse behavior, supporting interpretability, auditing, and model diagnostics.

1 INTRODUCTION

While modern deep time series forecasting models achieve strong predictive performance, their behavior remains difficult to interpret Kim et al. (2025), particularly from an inverse perspective: given a desired alternative forecast, it is unclear which temporal inputs are critical for reaching it and how they would need to change. Understanding this target-conditioned inverse behavior is crucial for supporting the diagnosis of target-dependent model behavior beyond retrospective attribution. When appropriate, feasibility constraints may also support decision-making. For example, they could help determine how an electricity demand profile would need to differ for the model to predict a lower future peak load. However, this requires solving a target-conditioned inverse forecasting problem: given a desired future trajectory, which input modifications would make a fixed forecaster predict it? The problem is nontrivial because, in multi-horizon forecasting, changes to the input can affect multiple forecast steps. Additionally, distinct patterns may yield the same trajectory, which makes inverse solutions nonunique and potentially unstable.

Existing explainability methods are largely retrospective Theissler et al. (2022), identifying which observed inputs contributed to a prediction through feature attribution, such as TimeSHAP Bento et al. (2021), WindowSHAP Nayebi et al. (2023), or TSHAP Nguyen & Ifrim (2026), attention-based interpretation Lim et al. (2020), or interpretable representation learning, such as

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.

EXCODER Hahn et al. (2026). However, they do not directly answer the inverse question of how the input must change to induce a desired future trajectory.

Counterfactual input explanations address this inverse perspective by identifying minimal input modifications that induce a desired change in prediction Wachter et al. (2018). Yet existing approaches typically optimize each instance independently Wang et al. (2023); Luo & Yin (2026); Wang et al. (2024), resulting in substantial test-time computational overhead and potentially inconsistent intervention patterns across similar inputs. Sparsity remains a central challenge for time-series counterfactuals Schlegel & Seidl (2026), which is particularly important for model-level interpretability, where minimal input interventions help isolate the timesteps a forecaster relies on to reach a specified alternative outcome. Furthermore, existing approaches often define targets relative to the input, for example, through trend shifts or input-dependent bounds Wang et al. (2023). While this improves feasibility, it couples the objective to the observed series and obscures the difficulty of achieving a predefined forecast.

Consequently, transparency into a model's target-conditioned inverse behavior in multi-horizon forecasting requires more than solving isolated counterfactual instances: it calls for a stable representation that can reveal target-dependent temporal relevance and distinguish it from the magnitude of the corresponding input-value changes. Unlike retrospective feature attribution, which explains a fixed prediction, this perspective identifies the modifications required to induce a specified future trajectory. Such a method should generalize across inputs, yield locally stable intervention patterns for similar samples under comparable targets, and reveal where and how the input must change to achieve the target.

To address these requirements:

- We reformulate counterfactual generation for multi-horizon forecasting as target-conditioned inverse forecasting, and empirically demonstrate that an amortized, forecaster-specific inverse mapping can replace repeated per-instance optimization in the evaluated settings.
- We introduce ConTex, a forecaster-architecture-agnostic input intervention architecture that extends counterfactual explanations beyond unified generation toward inspectable, structured interventions that separate temporal relevance from modification strength.
- We introduce target-conditioned intervention relevance, characterizing which input timesteps matter for inducing a specified desired forecast rather than explaining the observed prediction; ConTex operationalizes it through a learned temporal selection score.
- Across four forecasting backbones (PatchTST, N-HiTS, DLinear, TiDE) and four benchmark datasets (NN5, Electricity, Tourism, M4), ConTex achieves a favorable validity–minimality trade-off, attaining the highest average compactness across all settings while maintaining strong target validity and competitive proximity relative to methods with comparable validity, and reduces end-to-end generation costs by 12–36$\times$ compared to instance-wise optimization.

We view ConTex as a step toward extending counterfactual reasoning beyond sample generation by revealing target-dependent temporal dependencies at the model level.

2 Related Work

Counterfactual explanations were introduced by Wachter et al. (2018) as minimal input perturbations that change a model's prediction. In time series, counterfactual generation spans optimization-based, instance-based, and generative latent-space approaches Schlegel & Seidl (2026). However, existing work has predominantly focused on classification rather than forecasting settings (e.g., Ateş et al. (2021); Delaney et al. (2021); Bahri et al. (2022); Yan & Wang (2023)).

In time series forecasting, two related but distinct problem settings can be identified: generating alternative future trajectories and generating counterfactual inputs for a fixed forecaster. The former focuses on plausible or desired future outcomes, whereas the latter asks how the model input must be modified for a fixed predictor to produce a specified target forecast. Probabilistic models such as DeepAR Salinas et al. (2020) generate multiple plausible future trajectories, while architectures such as the Temporal Fusion Transformer Lim et al. (2020) support scenario analysis by conditioning on known future covariates. Related approaches incorporate causal reasoning into forecasting models

Ferchichi et al. (2025), enabling "what-if" analyses. Another related line of work Zuin & Veloso (2024) models future time-series distributions through quantile regression and searches for trajectories satisfying desired outcome constraint. Although this improves statistical plausibility, it does not optimize towards minimal input changes for a fixed predictive model and therefore addresses a different problem setting.

Counterfactual input generation for time series forecasting has received substantially less attention. ForecastCF Wang et al. (2023) is among the first methods to explicitly address counterfactual inputs for time series forecasting by iteratively perturbing historical inputs to achieve desired trend changes. Subsequent works extend this optimization-based paradigm to domains such as cryptocurrency forecasting Luo & Yin (2026) and multivariate settings Wang et al. (2024).

Amortization has previously been used to replace instance-wise optimization, e.g., with conditional models Looveren et al. (2021) and sequential algorithmic recourse Verma et al. (2021). However, multi-horizon forecasting poses a distinct challenge, as a desired forecast trajectory induces a coupled continuous inverse problem over the historical input. We first show that this inverse mapping can be learned through GenCF, a problem-specific direct generator. ConTex, however, goes beyond amortization by imposing an interpretable structure on the learned inverse mapping of the model's target-conditioned inverse behavior. While a unified counterfactual sample primarily captures target attainment, ConTex addresses this limitation by disentangling target-relative temporal relevance from the magnitude of the required input-value changes. We further hypothesize that this factorization supports identifying more compact patterns by modulating modification strength through an explicitly learned selection score.

3 Problem Formulation

We define a desired forecast as a trajectory within a set of bounds: $\mathcal{Y}_{\text{target}} = \left\{ y \in \mathbb{R}^H \;\middle|\; L_t \leq y_t \leq U_t,\ \forall t \in \{1, \ldots, H\} \right\}$ where $L \in \mathbb{R}^H$ and $U \in \mathbb{R}^H$ denote the lower and upper target bounds over horizon H. Given an univariate time series $x \in \mathbb{R}^{T_{\text{in}}}$, where T_{in} denotes the input sequence length, and a forecasting model $f_\theta : \mathbb{R}^{T_{\text{in}}} \to \mathbb{R}^H$ with fixed parameters and differentiable access, the counterfactual problem consists of finding a minimally modified input x' such that the resulting forecast satisfies the target constraint while preserving as much of x as possible: $x^{\text{cf}} = \arg\min_{x'} \mathcal{R}(x', x) \quad \text{s.t.} \quad f_\theta(x') \in \mathcal{Y}_{\text{target}}$. Here, preservation is measured by $\mathcal{R}(x', x) = \|x' - x\| + \frac{1}{T_{\text{in}}} \sum_{t=1}^{T_{\text{in}}} \mathbf{1}[|x_t - x'_t| > \tau]$, where the first term penalizes the scale of deviations from the original input, while the second penalizes the number of altered timesteps. This formulation can be interpreted as a *local constrained inverse problem*: rather than inverting f_θ globally, we seek an input x' in the vicinity of x whose prediction $f_\theta(x')$ satisfies the desired $\mathcal{Y}_{\text{target}}$. We further simplify this problem by expressing the counterfactual as an additive modification of the original input, $x^{\text{cf}} = x + z$, where $z \in \mathbb{R}^{T_{\text{in}}}$ denotes an intervention. This allows us to reformulate the problem as $z^* = \arg\min_z \mathcal{R}(x + z, x) \quad \text{s.t.} \quad f_\theta(x + z) \in \mathcal{Y}_{\text{target}}$. While pure amortization optimizes z directly, we seek a solution that enables distinction into temporal relevance and modification strength. Accordingly, we decompose into (i) whether a modification is required and (ii) how strongly it should be applied. To identify which timesteps are most important for target achievement, we use a soft temporal gating mechanism that assigns each timestep a bounded selection score $m_t \in [0, 1]$. This score serves as a learned proxy for target-conditioned intervention relevance and is regularized toward near-binary values to encourage a clear separation between selected and non-selected timesteps. Given the input x, its current prediction $\hat{y} = f(x)$, and target c, G_θ predicts $(m, s) = G_\theta(x, \hat{y}, c)$, where $s \in \mathbb{R}^{T_{\text{in}}}$ denotes the signed modification strength, and the final intervention is obtained as $z = m \odot s$.

4 Method

4.1 ConTex Architecture

Addressing the formulated problem requires a representation that captures temporal context, sequence position, and local input values while being explicitly conditioned on a continuous multi-horizon target trajectory. These requirements introduce multiple interdependent sources of complexity. A naive approach would attempt to model all of these aspects within a single monolithic component.

However, this quickly becomes problematic, as each of these subtasks is structurally distinct. We therefore follow a modular design principle, where each component is responsible for a single specified function. In particular, we separate (i) the extraction of temporal context h_x from the input sequence and (ii) the encoding of the desired target condition. These representations are combined through feature-wise linear modulation $\tilde{h}_x = h_x \odot (1 + \gamma) + \beta$, enabling the model to adapt its behavior to different desired trajectories. Additionally, the condition embedding is tiled across time and concatenated to the temporal representation before the prediction heads.

Since the input exhibits temporal dependencies, the temporal encoder must capture both local and long-range patterns, which can be achieved using either recurrent or convolutional architectures. We evaluate both a bidirectional LSTM (BiLSTM) Graves & Schmidhuber (2005) and a temporal convolutional network (TCN) Bai et al. (2018), with the TCN performing better on average. This choice is also aligned with our objective of producing sparse and localized modifications, as temporal convolutions impose a finite receptive field and thus introduce an inductive bias toward temporally localized intervention patterns. In contrast, the target condition is encoded by a multilayer perceptron (MLP) that maps a fixed-dimensional representation c into a latent embedding. For trend targets, c comprises trajectory center, width, current forecast, and slope, while arbitrary target patterns can be provided directly as sequences to the same encoder. For full implementation details, see A.1.1.

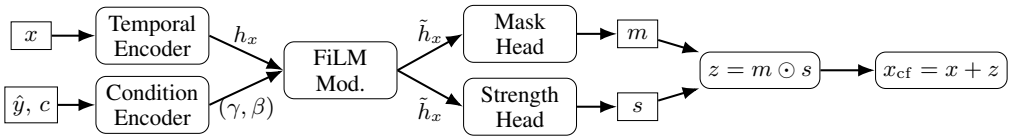

Figure 1: ConTex architecture. FiLM-based target conditioning (Perez et al. (2017)) modulates temporal features, from which a temporal selection score $m \in [0, 1]^T$ and signed modification strength $s \in \mathbb{R}^T$ are predicted. Their product $z = m \odot s$ defines the intervention added to the input to obtain the counterfactual sequence.

4.2 Training Objective

Counterfactual generation involves a trade-off between achieving the desired forecast behavior and minimizing the magnitude and extent of the intervention. We therefore optimize a composite objective consisting of four components.

Forecast alignment. To achieve high validity, we need to ensure that (i) a high percentage of timesteps satisfy the bounds, and (ii) the forecast accurately follows the desired trend. Accordingly, we penalize deviations from the midpoint of the target bounds as well as the (smooth) fraction of timesteps violating the constraints. Let $\hat{y}^{\text{cf}} = f(x^{\text{cf}})$ denote the forecast induced by the counterfactual input. We define $\mu_t = \frac{U_t + L_t}{2}, \quad w_t = \max\left(\frac{U_t - L_t}{2}, \varepsilon\right), \quad \delta_t = \frac{\hat{y}_t^{\text{cf}} - \mu_t}{w_t}$.

$$\mathcal{L}_{\text{center}} = \frac{1}{H} \sum_{t=1}^{H} |\delta_t|, \quad \mathcal{L}_{\text{valid}} = \frac{1}{H} \sum_{t=1}^{H} \sigma(\kappa(|\delta_t| - 1)). \tag{1}$$

Here, δ_t converts each constraint into a common coordinate system in which the boundary is at ± 1, to enable $\mathcal{L}_{\text{center}}$ to measure distance to the desired trajectory and $\mathcal{L}_{\text{valid}}$ to quantify violation frequency consistently across all timesteps. The steepness of the sigmoid transition is controlled by $\kappa > 0$.

Intervention regularization. To ensure minimal input interventions, both the magnitude of changes and the number of modified timesteps must be controlled. Accordingly, we impose input-space proximity and sparsity constraints addressing each objective:

$$\mathcal{L}_{\text{prox}} = \frac{1}{T_{\text{in}}} \sum_{t=1}^{T_{\text{in}}} \left| x_t^{\text{cf}} - x_t \right|, \quad \mathcal{L}_{\text{sparse}} = \frac{1}{T_{\text{in}}} \sum_{t=1}^{T_{\text{in}}} m_t + \frac{1}{T_{\text{in}}} \sum_{t=1}^{T_{\text{in}}} m_t(1 - m_t). \tag{2}$$

where T_{in} denotes the input sequence length. To avoid over-penalizing localized but necessary deviations, we use an L_1 proximity loss instead of L_2, promoting sparsity and interpretability

and discouraging diffuse small perturbations. Furthermore, to avoid degenerate solutions with uniformly distributed masks, $\mathcal{L}_{\text{sparse}}$ combines two terms: The first penalizes overall mask activation, encouraging sparse temporal selection, while the second pushes the selection scores towards near-binary values, reinforcing their role as a soft gate separate from modification strength. The final objective is then given by: $\mathcal{L}_{\text{total}} = \lambda_{\text{center}}\mathcal{L}_{\text{center}} + \lambda_{\text{valid}}\mathcal{L}_{\text{valid}} + \lambda_{\text{prox}}\mathcal{L}_{\text{prox}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}}$, balancing target satisfaction and minimal input modification.

5 Experiments

5.1 Experimental Setup

Datasets and forecasting models. We evaluate on four datasets spanning distinct temporal characteristics: M4 (daily, $T_{\text{in}} = 42$, $H = 14$; smooth and trend-dominated, Makridakis et al. (2020)), NN5 (daily, $T_{\text{in}} = 168$, $H = 56$; noisy and long-horizon, Andrawis et al. (2011)), Tourism (monthly, $T_{\text{in}} = 72$, $H = 24$; strongly seasonal, Godahewa et al. (2020b)), and Electricity (hourly, $T_{\text{in}} = 288$, $H = 96$; high-frequency, Godahewa et al. (2020a)). We further evaluate four forecasting backbones spanning distinct architectural paradigms: decomposition-based DLinear Zeng et al. (2022), hierarchical MLP-based N-HiTS Challu et al. (2022), encoder–decoder TiDE Das et al. (2024), and transformer-based PatchTST Nie et al. (2023).

Training protocol. For every dataset, all forecasting models are independently optimized, frozen, and reused across all counterfactual methods to ensure consistency and fairness. The resulting forecasting performance is reported in the appendix, Table 15. Architectural hyperparameters of ConTex are tuned once on the NN5 dataset, a challenging setting with long horizons and complex dynamics, and kept fixed across all other datasets. Only sparsity and proximity weights, learning rate, and dropout are tuned per dataset on the forecasting backbone with the lowest sMAPE and then reused across all backbones (see A.1).

Baselines. We compare ConTex against six baselines representing distinct counterfactual generation paradigms. For gradient-based instance-wise generation, we consider three ForecastCF variants Wang et al. (2023): the original implementation (FCF-Orig), a tuned variant (FCF-Tuned), and a variant optimizing on validity only (FCF-Valid). The latter two were tuned based on an extensive optimization procedure (see Appendix A.2). We contribute GenCF as a direct realization of amortized inverse forecasting, serving both as a baseline and as a methodological control for isolating the effect of the temporal selection–strength decomposition. For heuristic generation, we adapt BaseShift Wang et al. (2023) to externally specified trajectories by additively shifting the input mean toward the target level. Finally, an adapted version of BaseNN Wang et al. (2023) represents retrieval-based generation, selecting the nearest neighbor that satisfies the target constraint to maximize validity for the target set. Although BaseNN does not generate input interventions, it provides a strong retrieval-based reference using training samples. Further details are provided in Appendix A.6.

Benchmark. To study the feasibility of the proposed formulation in a systematic and reproducible manner, we evaluate all methods on a controlled, domain-agnostic trend benchmark. Targets are horizontal and increasing trajectories sampled across dataset percentiles, providing a standardized benchmark across datasets and forecasting backbones with comparable difficulty (see Appendix A.3.1). This controlled benchmark is designed to isolate the inverse forecasting problem from domain-specific feasibility and actionability assumptions to the model-level; full calibration details are provided in Appendix A.3. Importantly, to verify that the proposed formulation remains effective beyond standardized trend targets, we additionally retrain and evaluate on realistic, domain-specific target trajectories sampled from the corresponding data distributions.

Evaluation Metrics. We evaluate counterfactual quality using validity and minimality metrics following Wang et al. (2023). While the Validity Ratio measures stepwise target satisfaction, we additionally report Full Target Satisfaction (FTS), measuring the fraction of samples whose entire forecast satisfies the target. Since closeness and sparsity alone can be misleadingly favorable when modifications fail to achieve the desired target, we additionally report an Intervention Efficiency Rank (IER). For sufficiently valid configurations ($V \geq 0.5$), methods are ranked by $S = \frac{P}{V \cdot C}$ within each dataset–backbone setting, and IER denotes the average rank across datasets. Full definitions in A.4.

5.2 COUNTERFACTUAL BENCHMARK

Model	CF Model	NN5		Electricity		Tourism		M4	
		Ratio ↑	S-AUC ↑	Ratio ↑	S-AUC ↑	Ratio ↑	S-AUC ↑	Ratio ↑	S-AUC ↑
PatchTST	FCF-Tuned	0.332±0.024	0.083±0.011	0.657±0.017	0.364±0.010	0.482±0.023	0.365±0.027	0.697±0.091	0.679±0.088
	FCF-Orig	0.309±0.006	0.065±0.003	0.362±0.061	0.258±0.053	0.317±0.004	0.265±0.005	0.674±0.003	0.666±0.004
	FCF-Valid	0.699±0.039	0.387±0.040	0.704±0.044	0.291±0.020	0.672±0.023	0.468±0.018	0.957±0.016	0.954±0.016
	ConTex	0.863±0.015	0.784±0.038	0.923±0.000	0.729±0.023	**0.912**±0.012	**0.863**±0.019	**0.998**±0.000	**0.998**±0.000
	GenCF	0.893±0.024	**0.834**±0.014	**0.956**±0.013	**0.866**±0.022	**0.908**±0.010	**0.863**±0.016	**0.996**±0.001	**0.996**±0.001
	BaseShift	0.602	0.178	0.811	0.574	0.676	0.548	**0.996**	**0.996**
	BaseNN	**0.918**	0.770	0.829	0.553	0.863	0.754	**1.000**	**1.000**
N-HiTS	FCF-Tuned	0.538±0.054	0.214±0.046	0.738±0.032	0.470±0.032	0.476±0.024	0.363±0.032	0.842±0.033	0.819±0.029
	FCF-Orig	0.309±0.002	0.081±0.000	0.315±0.003	0.231±0.003	0.288±0.005	0.259±0.003	0.727±0.000	0.706±0.019
	FCF-Valid	0.983±0.002	0.905±0.012	0.557±0.016	0.410±0.017	0.735±0.014	0.587±0.031	**0.992**±0.000	**0.991**±0.000
	ConTex	**0.994**±0.008	0.966±0.048	0.821±0.009	0.540±0.030	**0.875**±0.036	**0.780**±0.064	**0.998**±0.000	**0.998**±0.000
	GenCF	**1.000**±0.000	**1.000**±0.000	0.822±0.043	0.534±0.052	**0.869**±0.015	0.766±0.059	**0.998**±0.000	**0.998**±0.000
	BaseShift	0.648	0.208	0.827	0.573	0.681	0.536	**0.996**	**0.995**
	BaseNN	0.914	0.689	**0.884**	**0.708**	0.851	0.737	**1.000**	**1.000**
DLinear	FCF-Tuned	0.399±0.010	0.120±0.006	0.664±0.021	0.339±0.028	0.381±0.003	0.266±0.002	0.832±0.008	0.782±0.027
	FCF-Orig	0.295±0.000	0.072±0.000	0.355±0.010	0.228±0.006	0.273±0.002	0.224±0.002	0.717±0.000	0.693±0.005
	FCF-Valid	**0.992**±0.000	0.919±0.004	0.786±0.022	0.282±0.008	0.776±0.014	0.533±0.020	0.716±0.004	0.705±0.002
	ConTex	**0.998**±0.000	**0.989**±0.003	0.939±0.012	0.685±0.051	**0.914**±0.017	0.860±0.020	**0.998**±0.000	**0.998**±0.000
	GenCF	**0.999**±0.000	**0.995**±0.004	**0.971**±0.010	**0.943**±0.005	**0.923**±0.008	**0.881**±0.012	**0.999**±0.000	**0.999**±0.000
	BaseShift	0.623	0.181	0.779	0.501	0.579	0.427	**0.995**	**0.995**
	BaseNN	0.894	0.579	0.824	0.544	0.839	0.705	**1.000**	**1.000**
TiDE	FCF-Tuned	0.625±0.041	0.213±0.036	0.685±0.023	0.359±0.025	0.493±0.017	0.375±0.005	0.831±0.008	0.808±0.016
	FCF-Orig	0.303±0.000	0.075±0.000	0.296±0.008	0.189±0.012	0.304±0.098	0.251±0.084	0.820±0.015	0.807±0.011
	FCF-Valid	**0.992**±0.000	0.928±0.002	0.758±0.013	0.272±0.020	0.771±0.004	0.515±0.015	0.811±0.068	0.785±0.065
	ConTex	**0.996**±0.002	0.977±0.009	**0.951**±0.008	0.786±0.064	**0.911**±0.013	**0.861**±0.022	**0.999**±0.000	**0.999**±0.000
	GenCF	**0.997**±0.002	**0.989**±0.006	**0.957**±0.011	**0.913**±0.021	**0.914**±0.006	**0.865**±0.009	**0.999**±0.000	**0.999**±0.000
	BaseShift	0.654	0.201	0.805	0.569	0.675	0.548	**0.995**	**0.995**
	BaseNN	0.860	0.447	0.829	0.570	0.840	0.726	**1.000**	**1.000**

Table 1: Counterfactual validity results averaged over five seeds. Values within 1% of the best are marked in bold. Corresponding Full Target Satisfaction results are reported in Appendix A.5.

Model	CF Model	NN5		Electricity		Tourism		M4		
		Comp. ↑	Prox. ↓	Comp. ↑	Prox. ↓	Comp. ↑	Prox. ↓	Comp. ↑	Prox. ↓	IER ↓
PatchTST	FCF-Tuned	0.46±0.06	2.46±0.67	0.23±0.00	**6.93**±0.82	0.30±0.02	8.01±1.51	0.87±0.03	0.28±0.08	4.75
	FCF-Orig	0.83±0.01	0.46±0.01	0.21±0.02	3.16±0.03	0.29±0.01	1.33±0.01	0.97±0.00	**0.25**±0.04	6.25
	FCF-Valid	0.10±0.02	**6.59**±0.74	0.05±0.01	43.39±6.84	0.06±0.00	20.15±0.83	0.40±0.02	2.04±0.17	4.50
	ConTex	**0.66**±0.07	9.21±0.53	**0.65**±0.05	33.81±1.19	**0.43**±0.04	15.57±1.03	**0.98**±0.00	0.74±0.00	**2.00**
	GenCF	0.30±0.06	9.67±0.83	0.34±0.03	42.16±1.83	0.24±0.01	**12.66**±0.43	0.97±0.00	1.21±0.02	3.00
	BaseShift	0.03	13.59	0.02	41.09	0.02	35.10	0.03	4.81	5.75
	BaseNN	0.03	17.10	0.06	34.99	0.12	28.39	0.79	0.51	3.50
N-HiTS	FCF-Tuned	0.36±0.02	**4.10**±0.25	0.25±0.00	**12.39**±0.75	0.31±0.03	7.50±1.62	0.69±0.01	0.37±0.02	3.25
	FCF-Orig	0.49±0.01	0.99±0.01	0.56±0.00	1.59±0.01	0.38±0.01	1.13±0.02	0.98±0.00	**0.27**±0.08	6.25
	FCF-Valid	0.03±0.00	10.66±0.07	0.06±0.00	32.43±0.42	0.10±0.01	15.40±1.04	0.37±0.00	2.22±0.01	4.50
	ConTex	**0.85**±0.02	19.28±1.45	**0.94**±0.00	45.17±2.39	**0.71**±0.03	15.58±1.70	**0.98**±0.00	0.74±0.00	**1.75**
	GenCF	0.69±0.02	16.12±1.24	0.61±0.07	35.96±1.67	0.40±0.03	**15.14**±1.53	0.98±0.00	0.75±0.00	3.00
	BaseShift	0.03	12.96	0.02	41.41	0.02	34.99	0.03	4.76	6.00
	BaseNN	0.04	14.61	0.06	36.76	0.13	28.67	0.60	0.83	4.50
DLinear	FCF-Tuned	0.46±0.01	5.17±0.08	0.23±0.01	**10.94**±1.02	0.43±0.03	10.58±0.42	0.76±0.01	0.61±0.01	4.75
	FCF-Orig	0.39±0.00	1.13±0.00	0.22±0.01	3.77±0.02	0.38±0.01	1.21±0.01	0.64±0.00	**0.55**±0.09	7.00
	FCF-Valid	0.01±0.00	14.61±0.01	0.04±0.00	49.28±3.42	0.06±0.00	23.10±0.57	0.35±0.00	2.90±0.03	5.00
	ConTex	**0.61**±0.02	20.95±0.13	**0.55**±0.03	37.74±1.00	**0.60**±0.01	16.31±1.07	**0.98**±0.00	0.76±0.00	**1.25**
	GenCF	0.52±0.01	19.69±0.16	0.08±0.00	31.91±0.11	0.34±0.03	**14.99**±0.74	0.92±0.00	0.90±0.01	2.50
	BaseShift	0.03	**12.97**	0.02	41.31	0.02	34.98	0.03	4.76	5.50
	BaseNN	0.03	17.42	0.06	35.85	0.11	28.33	0.59	0.84	3.75
TiDE	FCF-Tuned	0.27±0.01	**3.52**±0.22	0.23±0.01	**10.65**±0.65	0.28±0.01	6.23±0.43	0.72±0.04	**0.33**±0.06	3.00
	FCF-Orig	0.38±0.00	1.14±0.00	0.26±0.02	2.56±0.06	0.35±0.12	1.23±0.20	0.81±0.01	0.46±0.00	6.50
	FCF-Valid	0.01±0.00	14.17±0.01	0.05±0.00	45.92±0.92	0.06±0.00	22.83±0.39	0.34±0.01	2.82±0.09	5.25
	ConTex	**0.64**±0.01	21.69±0.22	**0.61**±0.02	34.82±0.49	**0.47**±0.02	**13.82**±0.38	**0.98**±0.00	0.91±0.00	**1.75**
	GenCF	0.55±0.01	20.39±0.10	0.28±0.14	32.38±0.90	0.42±0.03	14.96±0.90	0.86±0.00	1.11±0.00	3.00
	BaseShift	0.03	13.19	0.02	41.58	0.02	35.14	0.03	4.78	5.75
	BaseNN	0.05	13.20	0.06	35.89	0.12	27.99	0.57	0.82	4.00

Table 2: Results for input-space minimality averaged over five seeds. Best values are marked in bold. Compactness and proximity comparisons include only methods achieving at least 50% validity (otherwise greyed out), since favorable minimality scores are not meaningful when the target is largely unmet. ConTex achieves the best IER across all backbones and the highest compactness in 16/16 cases, preserving approximately 72% of input timesteps on average, compared with 52% for GenCF and 43% for FCF-Tuned.

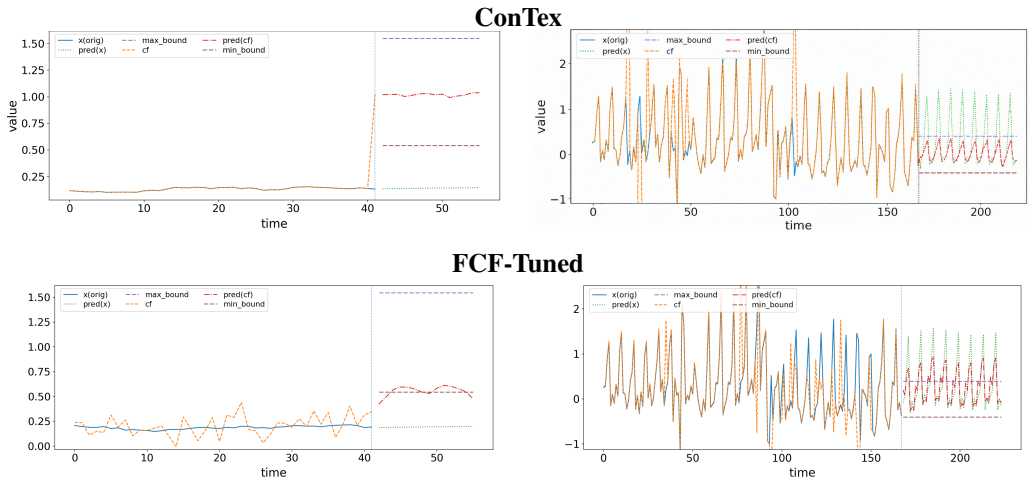

Figure 2: Across both low- and high-noise datasets, ConTex produces more localized and structured counterfactual explanations than FCF-Tuned. Especially on M4, modifications concentrate on the most recent timesteps, indicating that N-HiTS relies strongly on recent inputs. The more diffuse FCF-Tuned counterfactual makes these temporal drivers less distinct. This contrast is robust to the choice of τ: even when every non-zero perturbation is counted as a modification ($\tau = 0$), ConTex preserves a considerable fraction of the input, whereas FCF-Tuned perturbs nearly all timesteps (see Appendix A.7).

Table 1 reports the validity metrics of the generated counterfactuals. Learning the inverse forecasting mapping (GenCF & ConTex) enables consistently high target validity, outperforming instance-based FCF-Orig and FCF-Tuned across all dataset-backbone combinations. This is particularly notable relative to two strong references: FCF-Valid, which prioritizes target achievement over minimality, is outperformed in 13/16 settings, with comparable performance ($< 1\%$ difference) in the remaining 3/16; and BaseNN, which can retrieve highly valid trajectories from the observed data due to target calibration (see A.3.3), achieves the best Validity Ratio in only 2/16 cases and comparable performance in 4/16. These results indicate that a learned inverse mapping can replace per-instance optimization and match or exceed validity-only optimization while preserving counterfactual sparsity, which is analyzed in Table 2.

The learned inverse mappings also achieve strong compactness overall, with ConTex's relevance-strength decomposition yielding the highest average compactness across all settings, and, when relating to Intervention Efficiency (IER), reaches the highest efficiency across all backbones. We further observe that reaching both high validity and high sparsity is associated with increased proximity error. That is particularly pronounced on Electricity, where FCF-Valid improves average validity over FCF-Tuned by only 2.2% (0.686 to 0.701), while increasing average proximity more than fourfold (10.23 to 42.75). Compared with methods achieving at least 80% average validity (FCF-Valid, GenCF, and BaseNN), ConTex attains lower average proximity (17.94 vs. 18.70) while achieving substantially higher compactness. Moreover, this operating point can be shifted toward lower-proximity solutions through alternative proximity/sparsity regularization (see Appendix A.10) or alternative loss formulations (see 5.3), while retaining high compactness.

Another important aspect is how explanations vary for the same forecasting architecture when trained on datasets with different temporal characteristics. ConTex explanations indicate how the same forecasting architecture learns different temporal dependencies across datasets (Figure 2). This structure is further reflected in the learned temporal selection score, which serves as a proxy for target-conditioned intervention relevance by indicating which timesteps are selected for inducing the desired forecast. For the examples given above, we visualize the resulting scores in Figure 3.

We further validated the findings by retraining and testing on the realistic pattern benchmark, from which we show two examples for ConTex in Figure 4. Both represent challenging distributional shifts involving substantial differences in temporal structure, amplitude, and overall signal dynamics.

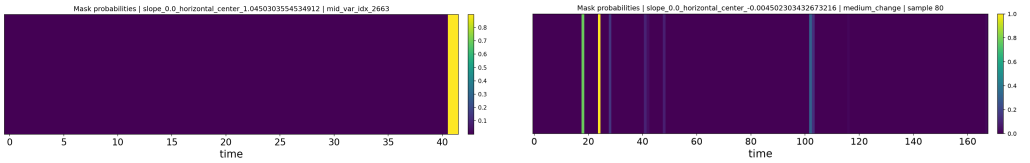

Figure 3: Interventional (target-conditioned) relevance: Learned temporal selection scores for an M4 sample (left) and an NN5 sample (right). Scores are strongly polarized with little intermediate activation, reflecting the intended soft-gating behavior. On M4, selection is concentrated on the most recent inputs, indicating a localized dependence of N-HiTS on recent history for inducing the target forecast; NN5 shows a broader temporal pattern.

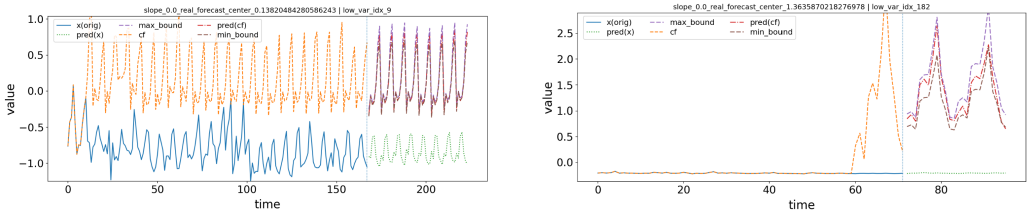

Figure 4: Despite substantial pattern shifts, ConTex retains highly competitive target validity and the best efficiency across all backbones, while improving input-space minimality relative to the trend benchmark, achieving the highest compactness in 13/16 and best proximity in 10/16 settings (see A.12)

Despite methodological differences, both ConTex and FCF-Tuned exhibit similar scaling behavior with increasing target difficulty (Figure 5). Harder targets generally require larger and less compact interventions, reflected by decreasing compactness and increasing proximity. However, while ConTex maintains stable validity across difficulty regimes, FCF-Tuned shows a substantial degradation in validity.

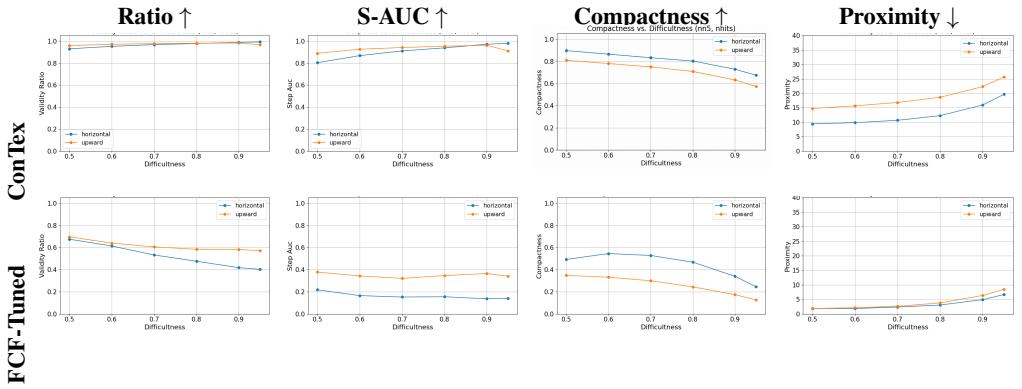

Figure 5: Metric behavior across increasing target difficulty on NN5 with N-HiTS.

Selection Faithfulness. To assess whether the learned temporal selection score faithfully identifies intervention-critical timesteps and thus serves as a meaningful proxy for relevance, we perform a deletion-based analysis using rankings by the selection score m, absolute modification strength, and final intervention $z = m \odot s$. If the mask successfully isolates intervention-critical relevance, deletion based on m should be substantially more disruptive than deletion based on s, while approaching the effect of deleting locations ranked by the complete effective intervention $z = m \odot s$. For the NN5 dataset with the N-HiTS model (e.g. in Figure 3, right), deleting only the top 5% mask-ranked locations reduces the validity ratio from 0.994 to 0.295, while ranking by modification strength yields a substantially smaller drop to 0.621; deleting low-ranked locations has little effect. Qualitatively similar trends occur across all datasets (see Appendix A.13).

Interventional Consistency. We assess local intervention consistency using the cosine similarity of intervention vectors among k nearest-neighbor inputs. Across all 64 configurations, ConTex achieves higher cosine similarity in 52 cases and improves the mean consistency from 0.776 to 0.891. Gains are particularly pronounced for N-HiTS, TiDE, and PatchTST. For example, on NN5 with N-HiTS, ConTex reaches 0.916 compared with 0.647 at $k = 1$. Full results across all configurations and five random seeds are provided in Appendix A.14.

Runtime Study. All experiments were conducted on an NVIDIA Tesla V100 GPU (32 GB VRAM) with CUDA 12.4 (driver 550.163.01). We report a representative study on NN5 using N-HiTS in the Appendix A.8. ConTex achieves a near-constant inference time of $\sim$0.007 s per sample. Even in the most challenging setting (95th percentile), this corresponds to a speedup of 3.7–4.2 orders of magnitude compared to instance-based generation via FCF-Tuned (108 s per sample), whose runtime increases with target difficulty. In the least challenging setting, ConTex remains several orders faster (e.g., 32.5 s vs. 0.007 s), demonstrating consistent gains across all difficulty levels. Accounting for training (Table 14), ConTex incurs a one-time cost of 5–70 minutes depending on the dataset and the configuration (10–20 minutes on M4, up to 60–70 minutes on NN5). In contrast, ForecastCF requires $\sim$18 hours for all NN5 settings across 5 seeds (3.75 days total vs. 5.8 hours), resulting in an overall reduction in computational cost of at least $12\times$–$36\times$.

5.3 Ablation Study

We evaluated each architectural design choice through five ablation studies, each averaged over five random seeds. Since the architecture was optimized on NN5 using N-HiTS, we use this configuration as a representative example (see A.11.1). The corresponding ablation results on Tourism (A.11.2) confirm the same trends. First, we evaluate the choice of an MAE proximity loss by replacing it with MSE $\mathcal{L}\mathrm{prox}$. Due to stronger penalties on large deviations, MSE discourages localized changes, reducing compactness and proximity (w/o $\mathcal{L}L1$). Secondly, we assess robustness to the optimization objective by removing ConTex's center and sparsity terms and instead using ForecastCF's margin-based validity and proximity objective (w/ FCF-Loss). This improves validity, S-AUC, and proximity while retaining high compactness, demonstrating that ConTex's performance is not dependent on its specific loss formulation. As a third ablation, we remove the conditioning encoder. This substantially impairs the model's ability to differentiate between desired target outcomes, leading to a substantial decrease in both validity ratio and S-AUC.

As a fourth ablation, we remove the decomposition into temporal selection and modification strength. This yields a modest increase in validity and reduces compactness by 15.7 percentage points (see A.11.1). This suggests that the decomposition provides more than interpretability. Explicitly representing intervention relevance separately from modification may help the model concentrate changes on fewer timesteps. As a final ablation, we remove $\hat{y}$ from the condition input. Interestingly, this does not adversely affect validity, but results in a slight reduction in compactness. This indicates that incorporating the current forecast is important for guiding the model toward higher temporal sparsity.

6 Conclusion

We demonstrate that the target-conditioned inverse behavior of a multi-horizon forecaster can be learned through a reusable model-level mapping, replacing repeated instance-wise optimization. Beyond that, we show that at comparable target validity, decomposition into temporal relevance and modification strength yields substantially more compact and interpretable model-level interventions while retaining competitive proximity and orders-of-magnitude faster generation. Moreover, deletion-based analysis shows that the relevance–strength decomposition yields a faithful selection-score proxy for timesteps critical to inducing a desired forecast. This provides a forward-looking, target-conditioned view of model behavior that complements retrospective attribution. With respect to system-level feasibility and plausibility, we deliberately study unconstrained model-level inverse behavior to interpret the forecaster itself, as feasibility constraints impose external assumptions on the intervention space and thereby alter the explanation. Extending ConTex to multivariate settings and, where desired, to constrained actionable intervention spaces is an important direction for future work. Additionally, ConTex's hyperparameters should be tuned on a per-dataset basis to achieve optimal performance.

AI USE STATEMENT

In this work, we used generative AI tools to assist with software development and debugging, including the implementation and refinement of experimental and evaluation code. We also used generative AI tools for language editing to improve the readability of the manuscript.

Generative AI tools were not used to autonomously formulate the research hypotheses, design the core methodology, or draw scientific conclusions from the experimental results. All AI-assisted code was reviewed and tested by the authors. The authors take full responsibility for the final content of this work, including all text, claims, code, and experimental results.

REPRODUCIBILITY STATEMENT

We provide all information required to reproduce our experiments. The model architecture, training objective, datasets, data splits, hyperparameters, baseline configurations, and evaluation metrics are documented in the main paper and appendix. All experiments are evaluated over five random seeds. Computational resources and runtimes are also reported. The forecasting backbones are based on the PyTorch implementations provided by the NeuralForecast library (Olivares et al., 2022). All datasets and forecasting architectures are publicly available. The complete codebase, including preprocessing, training, evaluation, configuration files, environment specifications, and reproduction instructions, is available in our anonymized repository at `https://anonymous.4open.science/r/ConTex`.

REFERENCES

Robert R. Andrawis, Amir F. Atiya, and Hisham El-Shishiny. Forecast combinations of computational intelligence and linear models for the nn5 time series forecasting competition. *International Journal of Forecasting*, 27(3):672–688, 2011. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2010.09.005. URL `https://www.sciencedirect.com/science/article/pii/S0169207010001445`. Special Section 1: Forecasting with Artificial Neural Networks and Computational Intelligence Special Section 2: Tourism Forecasting.

Emre Ateş, Burak Aksar, Vitus Leung, and Ayse Coskun. Counterfactual explanations for multivariate time series, 05 2021.

Omar Bahri, Soukaina Filali Boubrahimi, and Shah Muhammad Hamdi. Shapelet-based counterfactual explanations for multivariate time series, 2022. URL `https://arxiv.org/abs/2208.10462`.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL `https://arxiv.org/abs/1803.01271`.

João Bento, Pedro Saleiro, André F. Cruz, Mário A.T. Figueiredo, and Pedro Bizarro. Timeshap: Explaining recurrent models through sequence perturbations. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, KDD '21, pp. 2565–2573. ACM, August 2021. doi: 10.1145/3447548.3467166. URL `http://dx.doi.org/10.1145/3447548.3467166`.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting, 2022. URL `https://arxiv.org/abs/2201.12886`.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder, 2024. URL `https://arxiv.org/abs/2304.08424`.

Eoin Delaney, Derek Greene, and Mark T. Keane. Instance-based counterfactual explanations for time series classification, 2021. URL `https://arxiv.org/abs/2009.13211`.

A. Ferchichi, A.B. Abbes, V. Barra, and I.R. Farah. Trustworthy AI for Spatio-Temporal Forecasting via Counterfactual Causality, 2025.

Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. Electricity Hourly Dataset, June 2020a. URL https://zenodo.org/record/3889829.

Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. Tourism Monthly Dataset, June 2020b. URL https://zenodo.org/record/3889425.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL https://www.sciencedirect.com/science/article/pii/S0893608005001206. IJCNN 2005.

Yannik Hahn, Antonin Königsfeld, Hasan Tercan, and Tobias Meisen. Excoder: Explainable classification of discrete time series representations, 2026. URL https://arxiv.org/abs/2602.13087.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artificial Intelligence Review*, 58(7):216, April 2025. ISSN 1573-7462. doi: 10.1007/s10462-025-11223-9. URL https://link.springer.com/10.1007/s10462-025-11223-9.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting, 2020. URL https://arxiv.org/abs/1912.09363.

Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, and Oliver Cobb. Conditional generative models for counterfactual explanations, 2021. URL https://arxiv.org/abs/2101.10123.

X. Luo and W. Yin. Counterfactual Explanation-Based Cryptocurrency Price Prediction. *Entropy*, 28 (1), 2026. doi: 10.3390/e28010065. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-105028503537&doi=10.3390%2Fe28010065&partnerID=40&md5=76f11583c4b6477488a22a29f85fa8c4.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.04.014. URL https://www.sciencedirect.com/science/article/pii/S0169207019301128. M4 Competition.

Amin Nayebi, Sindhu Tipirneni, Chandan K Reddy, Brandon Foreman, and Vignesh Subbian. Windowshap: An efficient framework for explaining time-series classifiers based on shapley values, 2023. URL https://arxiv.org/abs/2211.06507.

Thach Le Nguyen and Georgiana Ifrim. Tshap: Fast and exact shap for explaining time series classification and regression. In Rita P. Ribeiro, Bernhard Pfahringer, Nathalie Japkowicz, Pedro Larrañaga, Alípio M. Jorge, Carlos Soares, Pedro H. Abreu, and João Gama (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track*, pp. 60–77, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-06078-5.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Kin G. Olivares, Cristian Challú, Azul Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/neuralforecast.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. URL https://arxiv.org/abs/1709.07871.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.07.001. URL https://www.sciencedirect.com/science/article/pii/S0169207019301888.

Udo Schlegel and Thomas Seidl. What-if explanations over time: Counterfactuals for time series classification, 2026. URL https://arxiv.org/abs/2603.27792.

Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. Explainable ai for time series classification: a review, taxonomy and research directions. *Ieee Access*, 10:100700–100724, 2022.

Sahil Verma, Keegan Hines, and John P. Dickerson. Amortized generation of sequential algorithmic recourses for black-box models, 2021. URL https://arxiv.org/abs/2106.03962.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr, 2018. URL https://arxiv.org/abs/1711.00399.

Z. Wang, I. Samsten, I. Miliou, and P. Papapetrou. COMET: Constrained Counterfactual Explanations for Patient Glucose Multivariate Forecasting, 2024.

Zhendong Wang, Ioanna Miliou, Isak Samsten, and Panagiotis Papapetrou. Counterfactual explanations for time series forecasting, 2023. URL https://arxiv.org/abs/2310.08137.

Jingquan Yan and Hao Wang. Self-interpretable time series prediction with counterfactual explanations, 2023. URL https://arxiv.org/abs/2306.06024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

G. Zuin and A. Veloso. Navigating Time's Possibilities: Plausible Counterfactual Explanations for Multivariate Time-Series Forecast through Genetic Algorithms, 2024.

A Appendix

A.1 Architecture Setup

A.1.1 General Configuration

Component	**Configuration**
Temporal Encoder	
Encoder Type	TCN
Encoding Dimension	64
Kernel Size	3
TCN Dropout	0.1
Receptive Field Ratio	1.0
Dilations	Exponentially increasing, 2^i
Condition Encoder	
Condition Input	Target parameters + forecast status
Conditioning MLP	2 layers, hidden dim 8, embed dim 16
Condition Injection	FiLM + temporal concatenation
FiLM Scaling	bounded, scale = 0.5
Prediction Heads	
Mask Head	Time-distributed MLP: 32–16–1, sigmoid
Strength Head	Time-distributed MLP: 32–16–1, tanh
Strength Scaling	Conditioned StrengthScaler, initial scale 1.0
Intervention	$z = m \odot s$
Output	$x^{\mathrm{cf}} = x + z$
Training	
Batch Size	128
Optimizer	Adam
Seeds	0–4
Loss Weights	
λ_{center}	0.2
λ_{valid}	1.0

Table 3: Architectural configuration of ConTex.

A.1.2 DATASET SPECIFIC

Dataset	**Learning Rate**	**TCN Dropout**	**Cond. Dropout**	λ_{sparse}	λ_{prox}
Electricity	2e-4	0.10	0.00	0.2	0.1
NN5	1e-4	0.05	0.00	0.35	0.3
M4	1e-3	0.05	0.05	0.2	0.2
Tourism	1e-4	0.10	0.00	0.4	0.25

Table 4: Dataset-specific hyperparameters for ConTex. A small set of optimization parameters is tuned for each dataset, while all other architectural components remain fixed.

Dataset	**Train**	**Validation**	**Test**
NN5	85% of training portion	10% of training portion	Final 56 days of each series
M4	85% of training portion	10% of training portion	Final 14 days of each series
Tourism	85% of training portion	10% of training portion	Final 24 months of each series
Electricity	85% of training portion	10% of training portion	Final 4 months of each series

Table 5: Dataset-specific data splits. Training and validation are constructed from the provided training portion of each dataset. The test evaluation is performed on the final forecasting horizon of each series.

Dataset	Resolution	Input Length	Horizon	Characteristics	Source
M4	Daily	42	14	Smooth, trend-dominated	Makridakis et al. (2020)
NN5	Daily	168	56	Noisy, long-horizon	Andrawis et al. (2011)
Tourism	Monthly	72	24	Strong seasonality	Godahewa et al. (2020b)
Electricity	Hourly	288	96	High-frequency	Godahewa et al. (2020a)

Table 6: Overview of datasets used in our experiments.

A.2 HYPERPARAMETER OPTIMIZATION: FORECASTCF ALGORITHM

To provide ForecastCF with a sufficient optimization budget, we follow a systematic per-dataset calibration protocol, illustrated exemplarily for FCF-Valid on NN5 in Figure 6. For each dataset, hyperparameters are calibrated on a disjoint set of 64 randomly selected samples. We progressively increase the learning rate and maximum iteration budget until further increases yield no meaningful improvement in target validity. The full search space is reported in Table 7, while the selected configurations for the ForecastCF algorithm per dataset are summarized in Table 8.

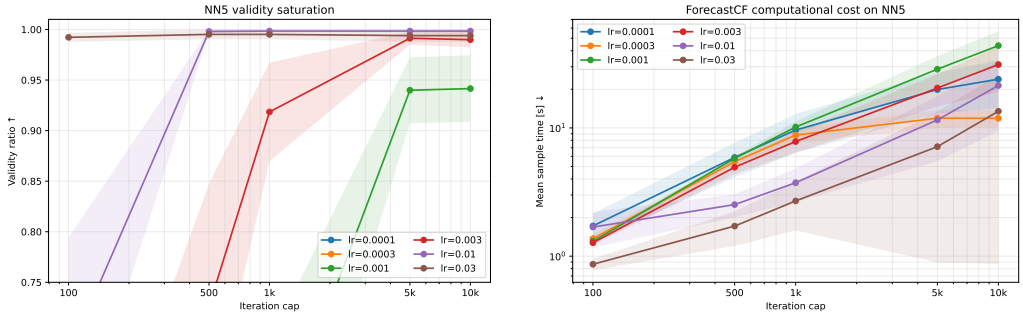

Figure 6: Optimization Protocol Example: The ForecastCF algorithm relies on test-time gradient optimization and is therefore sensitive to the learning rate and maximum optimization budget (iteration cap). We calibrate both parameters over a broad logarithmic range, extending the optimization budget up to $100\times$ that of the original implementation. Increasing the learning rate substantially accelerates convergence, however, the largest learning rate (3×10^{-2}) converges to a slightly lower validity level. We therefore select 10^{-2} for NN5, for which validity saturates around 500 iterations. To conservatively avoid under-optimization, we use the next larger budget of 1,000 iterations. Per-sample optimization is additionally terminated if the objective does not improve for 100 consecutive iterations, avoiding unnecessary computation after convergence.

Hyperparameter	**Search space**
Learning rate	$\{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}\}$
Maximum iteration budget	$\{100, 500, 1000, 5000, 10000\}$

Table 7: ForecastCF hyperparameter search space used for per-dataset calibration.

Dataset	**Learning rate**	**Iteration budget**
NN5	1×10^{-2}	1,000
Electricity	3×10^{-2}	5,000
Tourism	1×10^{-2}	10,000
M4	3×10^{-2}	500

Table 8: Dataset-specific ForecastCF parameters.

A.3 TARGET CALIBRATION

Controlled Target Family. To ensure a systematic and comparable evaluation across datasets, we use a controlled target family spanning the dataset distribution. We restrict the family to linear trajectories with tolerance, including constant and increasing trends, which capture fundamental level and trend shifts while remaining domain agnostic.

Difficulty Calibration. Counterfactual difficulty depends on both the width of the target interval and its position in the forecast distribution, with low-density targets generally being harder to achieve. We therefore calibrate targets using percentile-based intervals of the model output distribution. As a reference, we choose an interval for which a simple baseline attains approximately 50% validity (see Appendix A.3.2) and vary percentiles to control target difficulty. Increasing trends are sampled from upper percentiles of the slope distribution (e.g., the 90th percentile) at fixed interval width.

A.3.1 TARGET SAMPLING

Trend Target Sampling. Each trend target configuration is represented by its center, bound width δ, slope, and slope type. During training, configurations are sampled using validity-based stratification to expose the model to different levels of target difficulty. Specifically, configurations are categorized as easy ($V \geq 0.20$), medium ($0.05 \leq V < 0.20$), or hard ($V < 0.05$) based on their initial validity and sampled with probabilities 0.4, 0.4, and 0.2, respectively. During validation and testing, target configurations are cycled deterministically.

Realistic Target Sampling. To assess whether conclusions from the controlled benchmark extend to more complex target structures, we additionally retrain and evaluate using domain-specific target trajectories sampled from the corresponding data distributions (Appendix A.12). For these targets, the condition is represented directly by the full target trajectory rather than by the scalar trend parameters used above.

A.3.2 CALIBRATION EXAMPLE

Target	NN5	Electricity	Tourism	M4
0.5 horizontal	0.479	0.508	0.526	0.500
0.5 upward	0.268	0.220	0.145	0.489
0.6 horizontal	0.403	0.346	0.236	0.475
0.6 upward	0.212	0.152	0.092	0.475
0.7 horizontal	0.306	0.166	0.092	0.466
0.7 upward	0.159	0.089	0.054	0.464
0.8 horizontal	0.202	0.066	0.035	0.413
0.8 upward	0.107	0.046	0.027	0.408
0.9 horizontal	0.103	0.017	0.010	0.321
0.9 upward	0.049	0.014	0.009	0.315
0.95 horizontal	0.048	0.005	0.003	0.226
0.95 upward	0.020	0.005	0.002	0.220

Table 9: Example for calibration of target bounds across datasets on DLinear. The bounds width is adjusted so that forecasting purely based on existing dataset samples leads to approximately a 50 % validity ratio on horizontal bounds in the common data regime, ensuring feasibility on each dataset and ensuring that, at least in common regimes, samples already exist that satisfy the constraints to a certain degree.

A.3.3 FULL TARGET COVERAGE COMPARISON

	Electricity		M4	
Target	Ratio	Full	Ratio	Full
0.5 horizontal	0.500	0.292	0.500	0.493
0.5 upward	0.150	0.000	0.492	0.475
0.6 horizontal	0.354	0.099	0.477	0.470
0.6 upward	0.110	0.000	0.477	0.469
0.7 horizontal	0.178	0.012	0.468	0.461
0.7 upward	0.071	0.000	0.466	0.458
0.8 horizontal	0.067	0.000	0.412	0.405
0.8 upward	0.039	0.000	0.409	0.400
0.9 horizontal	0.019	0.000	0.318	0.312
0.9 upward	0.013	0.000	0.314	0.307
0.95 horizontal	0.005	0.000	0.221	0.215
0.95 upward	0.005	0.000	0.217	0.209

Table 10: Comparison for full target coverage for M4 and Electricity. While both were calibrated to a 50 percent validity ratio, the dataset distribution significantly differs, leading to a high percentage of samples that satisfy the target constraints on M4 with 100 % validity, and therefore ensuring a high mean validity of BaseNN, while only a few satisfy the corresponding target constraints on Electricity.

A.4 Evaluation Metrics

We evaluate counterfactual quality using validity and input-space minimality, adapted from prior work on time series counterfactuals Wang et al. (2023).

Validity. The *Validity Ratio* measures the fraction of forecasted timesteps satisfying the target constraints:

$$\text{ValidityRatio} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\left[\hat{y}'_{k,t} \in \mathcal{Y}_{\text{target},t}\right]\right), \tag{3}$$

where K is the number of counterfactuals and T the forecasting horizon. We further report *Stepwise Validity AUC*, capturing consecutive valid forecast steps.

Input-Space Minimality. To assess how much counterfactuals deviate from the original input, we use *Proximity* to measure the average Euclidean distance between the original input x and the counterfactual x', and *Compactness* to measure the proportion of timesteps that remain unchanged, where $\tau = 0.05$ denotes a tolerance in robust-scaled input space, such that negligible perturbations are treated as unchanged. We apply the same threshold to all methods.

$$\text{Proximity} = \frac{1}{K}\sum_{k=1}^{K}\|x_k - x'_k\|, \quad \text{Compactness} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\left[|x_{k,t} - x'_{k,t}| \leq \tau\right]\right). \tag{4}$$

Intervention Efficiency Rank (IER). Counterfactual quality is inherently multi-objective: an intervention should reach the target while remaining small and temporally localized. Although validity, proximity, and compactness characterize these properties individually, they do not directly reveal whether a favorable minimality score reflects an efficient intervention or an ineffective one. For example, leaving the input nearly unchanged yields excellent proximity and compactness but is not a meaningful counterfactual if the target is largely unmet. Conversely, high validity obtained through large or diffuse modifications indicates effective but inefficient intervention. We therefore introduce IER as a complementary summary of this trade-off. It does not replace the individual metrics, but facilitates comparison of how efficient the methods achieve valid and localized counterfactual outcomes. Let m denote a counterfactual method, b a forecasting backbone, and $d \in \mathcal{D}$ a dataset. For each sufficiently valid configuration ($V_{m,b,d} \geq 0.5$), we first compute $S_{m,b,d} = \frac{P_{m,b,d}}{V_{m,b,d} \cdot C_{m,b,d}}$, where P denotes proximity, V the Validity Ratio, and C compactness. Methods are then ranked in ascending order of S within each backbone–dataset setting. The Intervention Efficiency Rank is defined as $\text{IER}_{m,b} = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\text{rank}(S_{m,b,d})$, with lower values indicating better intervention efficiency. This ranking prevents datasets with numerically larger intervention costs from dominating the aggregate and makes IER reflect how consistently a method achieves a favorable validity–minimality trade-off across datasets. Configurations below the validity threshold receive a failure rank worse than every sufficiently valid method. Lower IER values therefore indicate more consistently efficient interventions.

Full Target Satisfaction. While the Validity Ratio provides a graded measure of target adherence by averaging over forecast steps, the counterfactual objective is fully satisfied only when the entire forecast trajectory lies within the target constraints. We therefore report *Full Target Satisfaction* (FTS), defined as the fraction of counterfactuals for which all forecast steps satisfy the target:

$$\text{FTS} = \frac{1}{K}\sum_{k=1}^{K}\mathbb{1}\left[V_k = 1\right], \tag{5}$$

where V_k denotes the fraction of forecast steps satisfying the target constraints for counterfactual sample k.

We report the corresponding results in Appendix A.5.

A.5 Full Target Satisfaction Results

Model	CF Model	NN5	Electricity	Tourism	M4
PatchTST	FCF-Tuned	0.001±0.002	0.096±0.010	0.143±0.055	0.524±0.031
	FCF-Orig	0.001±0.001	0.109±0.040	0.155±0.020	0.650±0.019
	FCF-Valid	0.069±0.025	0.065±0.003	0.199±0.035	0.946±0.021
	ConTex	0.238±0.200	0.317±0.061	0.793±0.040	**0.999**±0.000
	GenCF	0.372±0.144	**0.353**±0.107	**0.807**±0.065	**0.999**±0.000
	BaseShift	0.011±0.000	0.348±0.000	0.386±0.000	**0.995**±0.000
	BaseNN	**0.417**±0.000	0.333±0.000	0.667±0.000	**1.000**±0.000
N-HiTS	FCF-Tuned	0.013±0.009	0.138±0.036	0.142±0.044	0.502±0.015
	FCF-Orig	0.001±0.001	0.145±0.008	0.150±0.007	0.725±0.000
	FCF-Valid	0.568±0.048	0.273±0.015	0.280±0.060	**0.991**±0.000
	ConTex	**0.977**±0.010	0.433±0.013	**0.732**±0.084	**0.999**±0.000
	GenCF	**0.985**±0.023	0.432±0.006	0.423±0.006	**0.999**±0.000
	BaseShift	0.014±0.000	0.303±0.000	0.347±0.000	**0.995**±0.000
	BaseNN	0.417±0.000	**0.500**±0.000	0.667±0.000	**1.000**±0.000
DLinear	FCF-Tuned	0.000±0.001	0.098±0.027	0.060±0.002	0.549±0.012
	FCF-Orig	0.000±0.000	0.052±0.014	0.110±0.000	0.709±0.000
	FCF-Valid	0.698±0.014	0.066±0.015	0.218±0.014	0.658±0.001
	ConTex	**0.965**±0.000	**0.893**±0.022	**0.826**±0.023	**0.998**±0.000
	GenCF	**0.972**±0.019	0.750±0.214	0.800±0.009	**0.998**±0.000
	BaseShift	0.016±0.000	0.255±0.000	0.229±0.000	**0.995**±0.000
	BaseNN	0.167±0.000	0.333±0.000	0.591±0.000	**1.000**±0.000
TiDE	FCF-Tuned	0.007±0.008	0.117±0.019	0.191±0.007	0.461±0.005
	FCF-Orig	0.001±0.000	0.074±0.003	0.138±0.049	0.740±0.001
	FCF-Valid	0.719±0.014	0.074±0.008	0.162±0.005	0.717±0.057
	ConTex	**0.890**±0.012	**0.885**±0.021	**0.833**±0.013	**0.999**±0.000
	GenCF	0.883±0.071	**0.888**±0.035	0.807±0.019	**0.998**±0.000
	BaseShift	0.005±0.000	0.335±0.000	0.378±0.000	**0.995**±0.000
	BaseNN	0.000±0.000	0.250±0.000	0.667±0.000	**1.000**±0.000

Table 11: The FTS results provide a stricter view of the validity trends observed in the main benchmark, as a sample is counted as valid only if the entire forecast trajectory satisfies the target constraints. Consequently, FTS values are generally lower than the corresponding stepwise Validity Ratios. For ConTex and GenCF, across most backbone–dataset combinations, high Validity Ratios are nevertheless accompanied by high FTS values. In contrast, several baseline methods, despite reaching high stepwise validity, deteriorate more substantially under the stricter FTS criterion, indicating that isolated forecast-step violations remain more frequently and prevent full target satisfaction.

A.6 BASENN

Algorithm 1 BaseNN (adapted): Retrieval of samples satisfying the target trajectory. Candidate samples are selected based on constraint satisfaction, and the nearest neighbor is returned as the counterfactual. We set the minimum candidate count to $M = 10$ for all experiments.

Require: Training data x_{train}, model f_θ, bounds $[L, U]$, query x, minimum candidates M

1: Compute predictions $\hat{y}_i = f_\theta(x_i)$ for all $x_i \in x_{\text{train}}$

2: Compute validity scores:

$$v_i = \sum_{t=1}^{H} \mathbb{1}[\hat{y}_{i,t} \in [L_t, U_t]]$$

3: **for** $\tau \in \{H, H-1, \ldots, 0\}$ **do**

4: $\quad \mathcal{C}_\tau = \{x_i \mid v_i \geq \tau\}$

5: $\quad$ **if** $|\mathcal{C}_\tau| \geq M$ **then**

6: $\quad\quad$ **break**

7: $\quad$ **end if**

8: **end for**

9: **return** $x^{\text{cf}} = \arg\min_{x_i \in \mathcal{C}_\tau} \|x - x_i\|_2$

A.7 TAU SENSITIVITY

τ	ConTex Compactness ↑	FCF-Tuned Compactness ↑
0.000	0.414	0.000
0.010	0.813	0.083
0.025	0.827	0.1012
0.050	0.830	0.127
0.075	0.847	0.152
0.100	0.853	0.173

Table 12: Sensitivity analysis of counterfactual compactness to the modification threshold τ, on NN5 with N-HITS. ConTex consistently yields substantially higher compactness than FCF-Tuned across all evaluated thresholds. Even for $\tau = 0$, where every non-zero perturbation is counted as a modification, ConTex preserves a considerable fraction of input timesteps, whereas FCF-Tuned modifies virtually all locations with a non-zero value.

A.8 Runtime Results

A.8.1 NN5: Generation Time

Difficulty	Direction	Method	Mean	Min	Max	Total (s)
0.50	horizontal	ConTex	0.009688	0.006870	0.023669	0.8827
		FCF-Tuned	32.500600	7.210600	204.944300	3055.0566
	upward	ConTex	0.007094	0.006814	0.008191	0.6646
		FCF-Tuned	23.148600	7.205000	114.638600	2175.9652
0.60	horizontal	ConTex	0.007076	0.006819	0.007998	0.6633
		FCF-Tuned	15.892400	2.542100	209.665600	1493.8873
	upward	ConTex	0.007057	0.006855	0.007852	0.6618
		FCF-Tuned	26.798000	7.067000	80.611900	2519.0121
0.70	horizontal	ConTex	0.007076	0.006807	0.007839	0.6636
		FCF-Tuned	44.604500	6.970400	157.413000	4192.8264
	upward	ConTex	0.007049	0.006837	0.007913	0.6609
		FCF-Tuned	30.648600	6.848100	98.503000	2880.9678
0.80	horizontal	ConTex	0.007040	0.006831	0.007805	0.6603
		FCF-Tuned	60.288000	6.903700	186.274000	5667.0713
	upward	ConTex	0.007074	0.006832	0.007850	0.6634
		FCF-Tuned	50.014400	7.096800	121.303200	4701.3529
0.90	horizontal	ConTex	0.007056	0.006819	0.007823	0.6617
		FCF-Tuned	81.564700	11.216600	180.628700	7667.0773
	upward	ConTex	0.007078	0.006878	0.007855	0.6638
		FCF-Tuned	79.499700	8.407300	164.462500	7472.9678
0.95	horizontal	ConTex	0.007065	0.006856	0.007832	0.6626
		FCF-Tuned	108.473000	6.987100	266.547600	10196.4579
	upward	ConTex	0.007065	0.006857	0.007905	0.6624
		FCF-Tuned	104.345200	7.064400	223.090500	9808.4527

Table 13: Runtime statistics by difficulty percentile, direction, and method. For ConTex, inference time is measured as the forward-pass runtime, including synchronization via NumPy conversion to ensure that all GPU operations are completed.

A.8.2 ConTex Training Runtime Comparison

Component	**NN5 (s)**	**M4 (s)**
Forecast Model Preparation	3.23	22.01
Bounds Calibration	23.41	57.48
ConTex Training	**3074.57**	**314.58**
Validation	1248.20	106.02
Test Evaluation	3.42	67.19
Plot Generation	129.29	130.08
Total Pipeline	4542.80	758.34

Table 14: Runtime breakdown of the full ConTex pipeline with N-HiTS on NN5 in comparison to M4. While the total runtime includes all stages in the pipeline such as model preparation, bounds calibration, evaluation, and visualization, we focus on the core training time (highlighted in bold) for comparison, excluding validation time between epochs. While training time on ConTex in general is quite low, e.g. on M4 around 5-10 minutes, it increases with more complex and noisy data patterns such as on NN5 requiring approximately 58 minutes for the reported runtime study.

A.9 FORECASTING MODELS

Model	NN5	Electricity	Tourism	M4
PatchTST	22.363	**11.966**	19.213	3.022
N-HiTS	**22.015**	12.595	19.464	3.010
DLinear	22.428	15.192	**18.775**	**2.994**
TiDE	24.195	13.831	19.773	3.121

Table 15: Averaged sMAPE (%) for each model and dataset. The model implement

A.10 PROXIMITY SENSITIVITY

Sparsity	Proximity	Comp. ↑	Prox. ↓	S-AUC ↑	Ratio ↑
0.05	0.4	0.750 ± 0.008	14.679 ± 0.118	0.985 ± 0.008	0.997 ± 0.000
0.1	0.4	0.788 ± 0.017	15.888 ± 1.145	0.971 ± 0.025	0.997 ± 0.002
0.25	0.4	0.879 ± 0.018	18.624 ± 0.938	0.765 ± 0.124	0.944 ± 0.036
0.05	0.6	0.799 ± 0.003	15.258 ± 0.413	0.973 ± 0.010	0.996 ± 0.003
0.1	0.6	0.834 ± 0.006	15.470 ± 0.158	0.922 ± 0.026	0.983 ± 0.009
0.25	0.6	0.903 ± 0.020	17.351 ± 2.517	0.724 ± 0.244	0.922 ± 0.093

Table 16: Proximity sensitivity of ConTex across different regularization operating points. Even with weaker sparsity regularization, ConTex retains high compactness (approximately 75–80%) while achieving substantially lower proximity and high target validity. Increasing the sparsity weight further improves compactness, but generally requires larger modifications and can reduce target validity, consistent with the intervention being concentrated on fewer timesteps.

A.11 Ablation

A.11.1 NN5

Variant	Ratio ↑	S-AUC ↑	Comp. ↑	Prox. ↓
Contex Full	0.994 ± 0.008	0.966 ± 0.048	0.848 ± 0.016	19.283 ± 1.446
w/o $\mathcal{L}_{L1}$ (MSE)	0.956 ± 0.014	0.910 ± 0.022	0.141 ± 0.028	9.630 ± 0.144
w/ FCF-Loss	0.997 ± 0.002	0.986 ± 0.008	0.804 ± 0.002	17.011 ± 0.802
w/o Cond. Encoder	0.330 ± 0.012	0.092 ± 0.010	0.939 ± 0.017	11.171 ± 0.905
w/o Decomposition	1.000 ± 0.000	1.000 ± 0.000	0.691 ± 0.016	16.119 ± 1.243
w/o Status Forecast	0.997 ± 0.002	0.985 ± 0.006	0.844 ± 0.014	21.118 ± 0.698

Table 17: Ablation study on NN5 with N-HiTS. Values are reported as mean $\pm$ standard deviation across seeds.

A.11.2 Tourism

Variant	Ratio ↑	S-AUC ↑	Comp. ↑	Prox. ↓
Contex Full	0.875 ± 0.036	0.780 ± 0.064	0.713 ± 0.032	15.575 ± 1.702
w/o $\mathcal{L}_{L1}$ (MSE)	0.866 ± 0.016	0.793 ± 0.042	0.538 ± 0.089	8.501 ± 0.835
w/ FCF-Loss	0.831 ± 0.040	0.715 ± 0.057	0.723 ± 0.049	14.857 ± 0.849
w/o Cond. Encoder	0.183 ± 0.007	0.144 ± 0.008	0.927 ± 0.006	11.175 ± 1.033
w/o Decomposition	0.869 ± 0.015	0.766 ± 0.059	0.401 ± 0.029	15.144 ± 1.531
w/o Status Forecast	0.873 ± 0.021	0.812 ± 0.017	0.615 ± 0.025	14.158 ± 0.555

Table 18: Ablation study on the Tourism dataset using N-HiTS. Values are reported as mean $\pm$ standard deviation across random seeds. The main findings from NN5 are reproduced. Notably, the FCF-Loss variant further improves both compactness and proximity while maintaining comparable validity and S-AUC. This may partly result from removing the center-deviation term, as the margin-based objective no longer encourages additional input interventions once the target region is reached. Nevertheless, the high compactness achieved without explicit sparsity regularization suggests that the architectural decomposition itself contributes to temporal sparsity.

A.12 Realistic Pattern Validation

All methods are retrained and evaluated on the domain-specific target set.

Model	CF Model	NN5		Electricity		Tourism		M4	
		Ratio ↑	S-AUC ↑	Ratio ↑	S-AUC ↑	Ratio ↑	S-AUC ↑	Ratio ↑	S-AUC ↑
PatchTST	FCF-Tuned	0.047±0.006	0.014±0.002	0.428±0.013	0.092±0.006	0.192±0.005	0.099±0.002	0.129±0.047	0.116±0.042
	FCF-Orig	0.039±0.001	0.011±0.000	0.058±0.009	0.009±0.001	0.064±0.002	0.028±0.002	0.164±0.003	0.155±0.005
	FCF-Valid	0.112±0.008	0.024±0.001	0.582±0.057	0.103±0.021	0.488±0.014	0.170±0.009	0.197±0.040	0.150±0.033
	ConTex	0.036±0.007	0.014±0.001	0.512±0.108	0.086±0.026	**0.743**±0.028	**0.525**±0.031	0.848±0.024	0.819±0.022
	GenCF	0.051±0.041	0.017±0.006	0.716±0.019	0.205±0.027	0.668±0.032	0.500±0.024	0.799±0.019	0.776±0.025
	BaseShift	0.045	0.011	0.322	0.101	0.253	0.121	0.715	0.709
	BaseNN	**0.438**	**0.059**	**0.724**	**0.294**	0.576	0.265	**0.961**	**0.944**
N-HiTS	FCF-Tuned	0.070±0.005	0.019±0.001	0.397±0.027	0.081±0.005	0.191±0.006	0.097±0.002	0.341±0.020	0.284±0.017
	FCF-Orig	0.044±0.008	0.012±0.002	0.066±0.001	0.009±0.000	0.056±0.001	0.025±0.000	0.168±0.000	0.167±0.000
	FCF-Valid	0.585±0.009	0.102±0.001	0.573±0.009	0.081±0.004	0.442±0.029	0.194±0.025	0.293±0.003	0.270±0.002
	ConTex	0.593±0.042	0.113±0.016	0.527±0.058	0.070±0.006	0.475±0.019	0.244±0.020	0.833±0.040	0.829±0.037
	GenCF	**0.621**±0.034	**0.119**±0.012	0.575±0.014	0.089±0.009	0.470±0.016	0.257±0.026	0.805±0.021	0.800±0.020
	BaseShift	0.049	0.013	0.313	0.108	0.264	0.121	0.693	0.683
	BaseNN	0.443	0.075	**0.824**	**0.432**	**0.568**	**0.268**	**0.964**	**0.943**
DLinear	FCF-Tuned	0.056±0.000	0.019±0.000	0.349±0.024	0.060±0.006	0.144±0.003	0.071±0.002	0.532±0.006	0.324±0.030
	FCF-Orig	0.035±0.000	0.011±0.000	0.037±0.001	0.006±0.000	0.048±0.000	0.022±0.000	0.156±0.015	0.153±0.014
	FCF-Valid	0.708±0.006	0.162±0.004	0.672±0.006	0.152±0.005	0.570±0.015	0.247±0.010	0.297±0.021	0.177±0.017
	ConTex	0.544±0.034	0.093±0.008	0.743±0.018	0.208±0.012	**0.737**±0.025	**0.558**±0.028	0.813±0.022	0.802±0.026
	GenCF	**0.738**±0.047	**0.186**±0.038	**0.838**±0.026	**0.510**±0.047	0.729±0.047	0.551±0.045	0.805±0.018	0.798±0.019
	BaseShift	0.047	0.012	0.335	0.064	0.239	0.105	0.710	0.701
	BaseNN	0.393	0.058	0.669	0.167	0.569	0.238	**0.954**	**0.946**
TiDE	FCF-Tuned	0.076±0.002	0.022±0.001	0.371±0.013	0.067±0.005	0.193±0.015	0.096±0.002	0.358±0.007	0.300±0.003
	FCF-Orig	0.035±0.000	0.011±0.000	0.039±0.001	0.006±0.000	0.048±0.000	0.022±0.000	0.128±0.012	0.118±0.011
	FCF-Valid	0.706±0.001	0.162±0.004	0.673±0.014	0.148±0.013	0.559±0.001	0.240±0.002	0.282±0.070	0.230±0.051
	ConTex	**0.758**±0.029	**0.201**±0.034	0.667±0.041	0.179±0.041	**0.758**±0.021	**0.538**±0.022	0.833±0.061	0.822±0.061
	GenCF	0.751±0.020	0.190±0.017	**0.811**±0.045	**0.447**±0.053	0.648±0.042	0.476±0.043	0.792±0.012	0.787±0.013
	BaseShift	0.043	0.014	0.317	0.097	0.253	0.122	0.680	0.666
	BaseNN	0.359	0.051	0.677	0.228	0.553	0.228	**0.922**	**0.911**

Table 19: Benchmark results for counterfactual generation under realistic pattern matching. This task is substantially more challenging than trend matching, and validity decreases across all methods. Nevertheless, the overall findings remain consistent: the learned inverse forecasting methods, GenCF and ConTex, generally achieve higher validity than the instance-wise optimization approaches, FCF-Tuned and FCF-Orig. Similar to the trend benchmark, BaseNN and the validity-only optimized FCF-Valid remain strong reference methods; however, BaseNN is still outperformed in 8/16 cases. Considering ConTex alone, its relative performance also improves slightly compared with the trend benchmark, outperforming FCF-Valid in 11/16 cases.

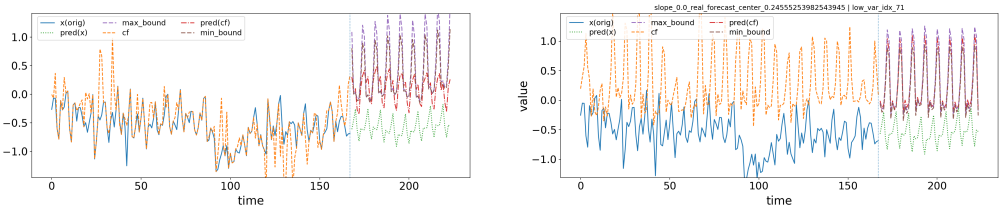

Figure 7: A qualitative example of counterfactuals generated for a target real-world pattern on the NN5 dataset using N-HiTS. FCF-Tuned (left) approximates the target region but fails to fully capture the required changes in amplitude and frequency. In gereral, we observe that ForecastCF produces more diffuse counterfactual modifications and forecasts that are only partially aligned with the desired target pattern. This limitation becomes more pronounced then in the trend benchmark, because the objective requires reproducing a complete temporal structure rather than merely reaching a target level or slope. ForecastCF often shifts the forecast toward the target region without accurately matching its local structure, whereas ConTex more closely reproduces properties such as peak locations, cycle frequency, and amplitude.

Model	CF Model	NN5		Electricity		Tourism		M4		
		Comp. ↑	Prox. ↓	Comp. ↑	Prox. ↓	Comp. ↑	Prox. ↓	Comp. ↑	Prox. ↓	IER ↓
PatchTST	FCF-Tuned	0.68±0.06	2.24±0.64	0.49±0.02	7.93±0.83	0.39±0.02	7.41±1.44	0.84±0.02	0.52±0.12	3.75
	FCF-Orig	0.90±0.02	0.35±0.04	0.41±0.02	2.17±0.04	0.37±0.01	1.22±0.01	0.98±0.00	0.31±0.00	3.75
	FCF-Valid	0.11±0.02	6.08±0.80	0.05±0.01	42.66±10.02	0.09±0.01	21.62±1.28	0.10±0.04	5.86±0.87	3.25
	ConTex	0.05±0.00	10.02±0.20	**0.68**±0.03	**21.77**±4.12	**0.23**±0.07	21.48±0.41	**0.90**±0.15	**1.41**±1.34	**1.00**
	GenCF	0.05±0.00	10.50±1.03	0.37±0.02	27.79±2.82	0.07±0.03	**20.74**±0.96	0.69±0.13	1.88±0.19	1.75
	BaseShift	0.07	6.30	0.02	39.46	0.03	31.69	0.03	4.64	3.50
	BaseNN	0.05	13.13	0.03	38.04	0.09	27.82	0.20	2.77	2.75
N-HiTS	FCF-Tuned	0.66±0.01	4.11±0.09	0.55±0.02	12.80±0.50	0.42±0.01	7.63±1.30	0.69±0.03	0.63±0.07	4.00
	FCF-Orig	0.65±0.08	0.76±0.12	0.62±0.02	1.38±0.05	0.42±0.01	1.06±0.01	0.98±0.00	0.32±0.00	4.00
	FCF-Valid	0.03±0.00	**12.50**±0.07	0.03±0.00	37.05±0.67	0.04±0.01	19.07±3.90	0.08±0.00	5.58±0.05	3.25
	ConTex	**0.06**±0.02	16.78±0.81	**0.83**±0.03	**21.44**±1.11	0.32±0.04	24.16±1.66	**0.98**±0.00	**0.75**±0.00	**1.25**
	GenCF	0.03±0.00	17.46±0.65	0.22±0.01	24.95±0.78	0.09±0.01	21.03±2.41	0.95±0.01	0.93±0.05	2.25
	BaseShift	0.07	6.30	0.02	39.46	0.03	31.69	0.03	4.63	3.75
	BaseNN	0.04	13.86	0.03	39.51	**0.08**	**27.48**	0.20	2.84	2.75
DLinear	FCF-Tuned	0.71±0.00	5.73±0.28	0.51±0.01	11.19±0.34	0.50±0.01	9.30±0.38	**0.96**±0.01	**0.73**±0.01	4.00
	FCF-Orig	0.58±0.00	0.87±0.00	0.40±0.01	2.22±0.03	0.43±0.01	1.14±0.01	0.79±0.00	0.61±0.01	5.00
	FCF-Valid	0.02±0.00	15.12±0.02	0.05±0.00	40.90±2.66	0.07±0.00	21.64±0.21	0.16±0.02	2.19±0.24	3.75
	ConTex	**0.09**±0.01	**13.46**±0.32	**0.37**±0.01	**26.66**±0.36	**0.27**±0.02	**20.52**±0.72	0.94±0.00	0.96±0.01	**1.25**
	GenCF	0.09±0.01	15.17±0.17	0.07±0.00	30.70±1.38	0.11±0.02	20.91±0.41	0.85±0.01	0.97±0.06	2.00
	BaseShift	0.07	6.30	0.02	39.46	0.03	31.69	0.03	4.64	4.75
	BaseNN	0.04	13.64	0.03	37.54	0.09	28.68	0.22	2.56	4.00
TiDE	FCF-Tuned	0.57±0.00	3.69±0.26	0.52±0.01	10.99±0.45	0.37±0.00	6.09±0.33	0.69±0.01	0.52±0.02	4.75
	FCF-Orig	0.57±0.00	0.88±0.00	0.39±0.01	2.24±0.01	0.41±0.00	1.17±0.00	0.83±0.01	0.63±0.01	4.75
	FCF-Valid	0.02±0.00	14.85±0.02	0.05±0.00	37.87±1.30	0.07±0.00	20.78±0.36	0.11±0.01	3.28±0.12	3.50
	ConTex	**0.10**±0.00	**14.68**±0.21	**0.45**±0.02	**23.52**±0.37	**0.14**±0.05	20.12±0.27	**0.90**±0.02	**1.15**±0.02	**1.00**
	GenCF	0.08±0.00	15.23±0.17	0.09±0.00	27.42±0.43	0.12±0.02	**19.38**±0.51	0.78±0.04	1.21±0.03	2.00
	BaseShift	0.07	6.30	0.02	39.46	0.03	31.69	0.03	4.63	4.50
	BaseNN	0.04	13.47	0.03	37.76	0.09	27.82	0.19	2.87	3.75

Table 20: Results for counterfactual minimality averaged over five seeds. Best values are marked in bold. Compactness and proximity comparisons include only methods achieving at least 50% validity, since favorable minimality scores are not meaningful when the target is largely unmet. Both ConTex and GenCF achieve the first and second best intervention efficiency ranking. Furthermore, ConTex remains competitive not only in temporal sparsity but also improves in proximity in relation to the trend benchmark, achieving the highest compactness in 13/16 and best proximity 10/16 settings.

A.13 SELECTION FAITHFULLNESS

To assess whether the learned temporal selection score identifies intervention-critical timesteps, we conducted a progressive deletion experiment across all datasets. For each sample, we ranked temporal locations by selection score m, absolute modification strength $|s|$, and final intervention $|m \odot s|$, and zeroed out at increasing fractions. Since ConTex constructs the final intervention as $z = m \odot s$, a meaningful decomposition should yield mask-based rankings that are substantially more informative than strength alone while remaining close to the ranking induced by the complete intervention. We further consider Bottom Deletion, which removes locations with the lowest mask values, and Random Deletion. A faithful mask should therefore cause substantially stronger degradation under top deletion than under bottom or random deletion.

Deletion strategy	Fraction	Ratio ↑	S-AUC ↑
Full intervention	–	0.994 ± 0.008	0.966 ± 0.048
Top mask deletion	5%	0.295 ± 0.016	0.050 ± 0.006
Top strength deletion	5%	0.621 ± 0.096	0.367 ± 0.094
Top intervention deletion	5%	0.337 ± 0.027	0.057 ± 0.008
Random deletion	5%	0.937 ± 0.002	0.822 ± 0.012
Bottom deletion	5%	0.994 ± 0.000	0.966 ± 0.000
Top mask deletion	10%	0.249 ± 0.004	0.039 ± 0.001
Top strength deletion	10%	0.499 ± 0.065	0.191 ± 0.072
Top intervention deletion	10%	0.256 ± 0.004	0.039 ± 0.001
Random deletion	10%	0.867 ± 0.004	0.681 ± 0.015
Bottom deletion	10%	0.994 ± 0.000	0.966 ± 0.000
Top mask deletion	20%	0.230 ± 0.004	0.035 ± 0.001
Top strength deletion	20%	0.382 ± 0.046	0.094 ± 0.017
Top intervention deletion	20%	0.231 ± 0.003	0.035 ± 0.001
Random deletion	20%	0.694 ± 0.010	0.434 ± 0.016
Bottom deletion	20%	0.994 ± 0.000	0.966 ± 0.000
Top mask deletion	40%	0.224 ± 0.001	0.034 ± 0.000
Top strength deletion	40%	0.290 ± 0.019	0.049 ± 0.005
Top intervention deletion	40%	0.225 ± 0.001	0.034 ± 0.000
Random deletion	40%	0.446 ± 0.015	0.173 ± 0.006
Bottom deletion	40%	0.994 ± 0.000	0.966 ± 0.000
Top mask deletion	60%	0.224 ± 0.001	0.034 ± 0.000
Top strength deletion	60%	0.256 ± 0.014	0.043 ± 0.005
Top intervention deletion	60%	0.224 ± 0.001	0.034 ± 0.000
Random deletion	60%	0.325 ± 0.007	0.078 ± 0.004
Bottom deletion	60%	0.994 ± 0.000	0.966 ± 0.000
Top mask deletion	80%	0.224 ± 0.001	0.034 ± 0.000
Top strength deletion	80%	0.247 ± 0.005	0.042 ± 0.004
Top intervention deletion	80%	0.224 ± 0.001	0.034 ± 0.000
Random deletion	80%	0.260 ± 0.003	0.044 ± 0.001
Bottom deletion	80%	0.978 ± 0.003	0.931 ± 0.007

Table 21: Mask-faithfulness results for NN5, averaged over all target configurations and 5 training seeds. Values are mean $\pm$ standard deviation across seeds. The deletion analysis supports a meaningful relevance–strength decomposition: deletion based on the selection score induces degradation comparable to deletion based on the complete intervention $|m \odot s|$, while ranking based on strength only $|s|$ is substantially less disruptive.

Deletion strategy	Fraction	Ratio ↑	S-AUC ↑
Full intervention	–	0.875 ± 0.035	0.780 ± 0.019
Top mask deletion	5%	0.169 ± 0.008	0.158 ± 0.007
Top strength deletion	5%	0.547 ± 0.049	0.492 ± 0.037
Top intervention deletion	5%	0.217 ± 0.005	0.188 ± 0.006
Random deletion	5%	0.766 ± 0.034	0.668 ± 0.025
Bottom deletion	5%	0.875 ± 0.035	0.779 ± 0.020
Top mask deletion	10%	0.160 ± 0.005	0.145 ± 0.004
Top strength deletion	10%	0.489 ± 0.034	0.449 ± 0.034
Top intervention deletion	10%	0.178 ± 0.004	0.154 ± 0.002
Random deletion	10%	0.672 ± 0.027	0.576 ± 0.019
Bottom deletion	10%	0.874 ± 0.035	0.779 ± 0.019
Top mask deletion	20%	0.153 ± 0.003	0.139 ± 0.003
Top strength deletion	20%	0.453 ± 0.029	0.406 ± 0.028
Top intervention deletion	20%	0.165 ± 0.003	0.145 ± 0.002
Random deletion	20%	0.545 ± 0.020	0.462 ± 0.013
Bottom deletion	20%	0.872 ± 0.035	0.775 ± 0.020
Top mask deletion	40%	0.152 ± 0.001	0.136 ± 0.001
Top strength deletion	40%	0.414 ± 0.019	0.361 ± 0.017
Top intervention deletion	40%	0.155 ± 0.001	0.136 ± 0.001
Random deletion	40%	0.385 ± 0.007	0.326 ± 0.009
Bottom deletion	40%	0.852 ± 0.036	0.754 ± 0.026
Top mask deletion	60%	0.152 ± 0.001	0.135 ± 0.001
Top strength deletion	60%	0.362 ± 0.012	0.314 ± 0.015
Top intervention deletion	60%	0.152 ± 0.001	0.135 ± 0.002
Random deletion	60%	0.277 ± 0.004	0.236 ± 0.005
Bottom deletion	60%	0.776 ± 0.037	0.671 ± 0.039
Top mask deletion	80%	0.152 ± 0.001	0.134 ± 0.002
Top strength deletion	80%	0.299 ± 0.016	0.259 ± 0.012
Top intervention deletion	80%	0.152 ± 0.001	0.134 ± 0.002
Random deletion	80%	0.207 ± 0.003	0.180 ± 0.003
Bottom deletion	80%	0.584 ± 0.025	0.494 ± 0.019

Table 22: Mask-faithfulness results for Tourism, averaged over all target configurations and 5 training seeds. Values are mean ± standard deviation across seeds.

Deletion strategy	Fraction	Ratio ↑	S-AUC ↑
Full intervention	–	0.821 ± 0.009	0.540 ± 0.030
Top mask deletion	5%	0.163 ± 0.000	0.054 ± 0.004
Top strength deletion	5%	0.661 ± 0.072	0.389 ± 0.062
Top intervention deletion	5%	0.165 ± 0.002	0.054 ± 0.003
Random deletion	5%	0.783 ± 0.016	0.509 ± 0.029
Bottom deletion	5%	0.821 ± 0.017	0.540 ± 0.033
Top mask deletion	10%	0.162 ± 0.001	0.054 ± 0.004
Top strength deletion	10%	0.636 ± 0.094	0.371 ± 0.076
Top intervention deletion	10%	0.162 ± 0.001	0.054 ± 0.004
Random deletion	10%	0.746 ± 0.014	0.478 ± 0.026
Bottom deletion	10%	0.821 ± 0.017	0.540 ± 0.033
Top mask deletion	20%	0.161 ± 0.001	0.054 ± 0.004
Top strength deletion	20%	0.596 ± 0.108	0.341 ± 0.082
Top intervention deletion	20%	0.161 ± 0.001	0.054 ± 0.004
Random deletion	20%	0.671 ± 0.011	0.422 ± 0.018
Bottom deletion	20%	0.821 ± 0.017	0.540 ± 0.033
Top mask deletion	40%	0.161 ± 0.002	0.054 ± 0.004
Top strength deletion	40%	0.286 ± 0.078	0.157 ± 0.076
Top intervention deletion	40%	0.161 ± 0.002	0.054 ± 0.004
Random deletion	40%	0.531 ± 0.007	0.315 ± 0.012
Bottom deletion	40%	0.821 ± 0.017	0.540 ± 0.033
Top mask deletion	60%	0.161 ± 0.002	0.054 ± 0.004
Top strength deletion	60%	0.161 ± 0.002	0.054 ± 0.004
Top intervention deletion	60%	0.161 ± 0.002	0.054 ± 0.004
Random deletion	60%	0.402 ± 0.002	0.222 ± 0.005
Bottom deletion	60%	0.821 ± 0.017	0.540 ± 0.033
Top mask deletion	80%	0.161 ± 0.002	0.054 ± 0.004
Top strength deletion	80%	0.161 ± 0.002	0.054 ± 0.004
Top intervention deletion	80%	0.161 ± 0.002	0.054 ± 0.004
Random deletion	80%	0.277 ± 0.002	0.134 ± 0.005
Bottom deletion	80%	0.821 ± 0.017	0.540 ± 0.033

Table 23: Mask-faithfulness results for Electricity, averaged over all target configurations and 5 training seeds. Values are mean $\pm$ standard deviation across seeds.

Deletion strategy	Fraction	Ratio ↑	S-AUC ↑
Full intervention	–	0.999 ± 0.000	0.999 ± 0.000
Top mask deletion	5%	0.344 ± 0.000	0.344 ± 0.000
Top strength deletion	5%	0.362 ± 0.037	0.362 ± 0.037
Top intervention deletion	5%	0.344 ± 0.000	0.344 ± 0.000
Random deletion	5%	0.952 ± 0.002	0.952 ± 0.002
Bottom deletion	5%	0.999 ± 0.000	0.999 ± 0.000
Top mask deletion	10%	0.344 ± 0.000	0.344 ± 0.000
Top strength deletion	10%	0.362 ± 0.037	0.362 ± 0.037
Top intervention deletion	10%	0.344 ± 0.000	0.344 ± 0.000
Random deletion	10%	0.921 ± 0.001	0.921 ± 0.001
Bottom deletion	10%	0.999 ± 0.000	0.999 ± 0.000
Top mask deletion	20%	0.344 ± 0.000	0.344 ± 0.000
Top strength deletion	20%	0.362 ± 0.037	0.362 ± 0.037
Top intervention deletion	20%	0.344 ± 0.000	0.344 ± 0.000
Random deletion	20%	0.858 ± 0.000	0.858 ± 0.000
Bottom deletion	20%	0.999 ± 0.000	0.999 ± 0.000
Top mask deletion	40%	0.344 ± 0.000	0.344 ± 0.000
Top strength deletion	40%	0.361 ± 0.035	0.361 ± 0.035
Top intervention deletion	40%	0.344 ± 0.000	0.344 ± 0.000
Random deletion	40%	0.735 ± 0.001	0.734 ± 0.001
Bottom deletion	40%	0.999 ± 0.000	0.999 ± 0.000
Top mask deletion	60%	0.344 ± 0.000	0.344 ± 0.000
Top strength deletion	60%	0.361 ± 0.034	0.361 ± 0.034
Top intervention deletion	60%	0.344 ± 0.000	0.344 ± 0.000
Random deletion	60%	0.593 ± 0.001	0.592 ± 0.001
Bottom deletion	60%	0.999 ± 0.000	0.999 ± 0.000
Top mask deletion	80%	0.344 ± 0.000	0.344 ± 0.000
Top strength deletion	80%	0.355 ± 0.022	0.355 ± 0.022
Top intervention deletion	80%	0.344 ± 0.000	0.344 ± 0.000
Random deletion	80%	0.469 ± 0.001	0.468 ± 0.001
Bottom deletion	80%	0.999 ± 0.000	0.999 ± 0.000

Table 24: Mask-faithfulness results for M4, averaged over all target configurations and 5 training seeds. Values are mean ± standard deviation across seeds.

A.14 INTERVENTIONAL CONSISTENCY

We evaluate local interventional consistency across forecasting models and datasets by comparing ConTex with the tuned gradient-based baseline FCF-Tuned. For a fixed target condition, nearby normalized inputs should receive similarly directed interventions. We therefore identify the k nearest neighbors of each input and measure the cosine similarity between their intervention vectors. Higher similarity indicates more consistent intervention directions and temporal patterns among similar inputs. While ConTex improves on most settings, the main exception is DLinear, for which FCF-Tuned is already highly consistent, particularly on M4 and Electricity. This may reflect DLinear's simple linear structure, which can yield strongly aligned optimization directions for nearby inputs.

Model	CF Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$
DLinear	FCF-Tuned	$0.892_{\pm 0.010}$	$0.879_{\pm 0.011}$	$0.872_{\pm 0.010}$	$0.861_{\pm 0.010}$
	ConTex	$0.912_{\pm 0.003}$	$0.881_{\pm 0.003}$	$0.847_{\pm 0.004}$	$0.766_{\pm 0.005}$
N-HiTS	FCF-Tuned	$0.647_{\pm 0.035}$	$0.633_{\pm 0.037}$	$0.628_{\pm 0.035}$	$0.607_{\pm 0.034}$
	ConTex	$0.916_{\pm 0.004}$	$0.886_{\pm 0.007}$	$0.857_{\pm 0.009}$	$0.805_{\pm 0.009}$
PatchTST	FCF-Tuned	$0.684_{\pm 0.023}$	$0.673_{\pm 0.022}$	$0.669_{\pm 0.021}$	$0.660_{\pm 0.019}$
	ConTex	$0.818_{\pm 0.012}$	$0.782_{\pm 0.012}$	$0.751_{\pm 0.012}$	$0.705_{\pm 0.012}$
TiDE	FCF-Tuned	$0.573_{\pm 0.028}$	$0.554_{\pm 0.029}$	$0.543_{\pm 0.028}$	$0.524_{\pm 0.028}$
	ConTex	$0.913_{\pm 0.004}$	$0.883_{\pm 0.004}$	$0.850_{\pm 0.005}$	$0.769_{\pm 0.006}$

Table 25: Neighborhood consistency by forecasting model (nn5). Values are mean cosine similarity across seeds with standard deviation across seeds.

Model	CF Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$
DLinear	FCF-Tuned	$0.888_{\pm 0.007}$	$0.867_{\pm 0.007}$	$0.855_{\pm 0.007}$	$0.836_{\pm 0.006}$
	ConTex	$0.844_{\pm 0.006}$	$0.808_{\pm 0.007}$	$0.783_{\pm 0.006}$	$0.726_{\pm 0.006}$
N-HiTS	FCF-Tuned	$0.781_{\pm 0.011}$	$0.757_{\pm 0.013}$	$0.742_{\pm 0.012}$	$0.718_{\pm 0.012}$
	ConTex	$0.860_{\pm 0.005}$	$0.831_{\pm 0.007}$	$0.813_{\pm 0.009}$	$0.782_{\pm 0.011}$
PatchTST	FCF-Tuned	$0.613_{\pm 0.081}$	$0.578_{\pm 0.081}$	$0.557_{\pm 0.081}$	$0.526_{\pm 0.083}$
	ConTex	$0.840_{\pm 0.007}$	$0.802_{\pm 0.008}$	$0.777_{\pm 0.009}$	$0.722_{\pm 0.011}$
TiDE	FCF-Tuned	$0.676_{\pm 0.023}$	$0.641_{\pm 0.027}$	$0.623_{\pm 0.027}$	$0.598_{\pm 0.029}$
	ConTex	$0.830_{\pm 0.006}$	$0.794_{\pm 0.006}$	$0.769_{\pm 0.007}$	$0.710_{\pm 0.007}$

Table 26: Neighborhood consistency by forecasting model (tourism). Values are mean cosine similarity across seeds with standard deviation across seeds.

Model	CF Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$
DLinear	FCF-Tuned	$0.974_{\pm 0.003}$	$0.971_{\pm 0.003}$	$0.970_{\pm 0.003}$	$0.964_{\pm 0.004}$
	ConTex	$0.987_{\pm 0.000}$	$0.984_{\pm 0.001}$	$0.972_{\pm 0.001}$	$0.946_{\pm 0.001}$
N-HiTS	FCF-Tuned	$0.893_{\pm 0.023}$	$0.887_{\pm 0.024}$	$0.883_{\pm 0.025}$	$0.872_{\pm 0.026}$
	ConTex	$0.965_{\pm 0.003}$	$0.957_{\pm 0.003}$	$0.946_{\pm 0.003}$	$0.928_{\pm 0.005}$
PatchTST	FCF-Tuned	$0.874_{\pm 0.014}$	$0.867_{\pm 0.015}$	$0.862_{\pm 0.015}$	$0.849_{\pm 0.016}$
	ConTex	$0.983_{\pm 0.002}$	$0.980_{\pm 0.003}$	$0.972_{\pm 0.004}$	$0.955_{\pm 0.006}$
TiDE	FCF-Tuned	$0.971_{\pm 0.003}$	$0.968_{\pm 0.003}$	$0.966_{\pm 0.003}$	$0.959_{\pm 0.003}$
	ConTex	$0.984_{\pm 0.000}$	$0.981_{\pm 0.000}$	$0.968_{\pm 0.000}$	$0.942_{\pm 0.000}$

Table 27: Neighborhood consistency by forecasting model (electricity). Values are mean cosine similarity across seeds with standard deviation across seeds.

Model	CF Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$
DLinear	FCF-Tuned	$0.986_{\pm 0.004}$	$0.984_{\pm 0.004}$	$0.983_{\pm 0.004}$	$0.981_{\pm 0.005}$
	ConTex	$0.979_{\pm 0.002}$	$0.974_{\pm 0.001}$	$0.971_{\pm 0.001}$	$0.964_{\pm 0.000}$
N-HiTS	FCF-Tuned	$0.827_{\pm 0.033}$	$0.809_{\pm 0.037}$	$0.799_{\pm 0.039}$	$0.785_{\pm 0.041}$
	ConTex	$0.980_{\pm 0.001}$	$0.974_{\pm 0.001}$	$0.971_{\pm 0.001}$	$0.964_{\pm 0.000}$
PatchTST	FCF-Tuned	$0.747_{\pm 0.024}$	$0.736_{\pm 0.024}$	$0.730_{\pm 0.024}$	$0.721_{\pm 0.024}$
	ConTex	$0.972_{\pm 0.001}$	$0.973_{\pm 0.001}$	$0.970_{\pm 0.000}$	$0.962_{\pm 0.000}$
TiDE	FCF-Tuned	$0.705_{\pm 0.045}$	$0.684_{\pm 0.043}$	$0.673_{\pm 0.042}$	$0.659_{\pm 0.042}$
	ConTex	$0.979_{\pm 0.001}$	$0.975_{\pm 0.001}$	$0.972_{\pm 0.001}$	$0.965_{\pm 0.001}$

Table 28: Neighborhood consistency by forecasting model (m4). Values are mean cosine similarity across seeds with standard deviation across seeds.