
How to Unlock Time Series Editing? Diffusion-Driven Approach with Multi-Grained Control

Hao Yu*
McGill University
hao.yu2@mail.mcgill.ca

Chu Xin Cheng*
California Institute of Technology
ccheng2@caltech.edu

Runlong Yu
University of Pittsburgh
ruy59@pitt.edu

Yuyang Ye
Rutgers University
yuyang.ye@rutgers.edu

Shiwei Tong†
IEGG, Tencent
shiweitong@tencent.com

Zhaofeng Liu
IEGG, Tencent
zhaofengliu@tencent.com

Defu Lian
University of Science and Technology of China
liandefu@ustc.edu.cn

Abstract

Recent advances in time series generation have shown promise, yet controlling properties in generated sequences remains challenging. Time Series Editing (TSE) — making precise modifications while preserving temporal coherence — current methods struggle to consider multi-grain controls, including both point-level constraints and segment-level controls. We introduce the COCKTAILEDIT framework to enable simultaneous, flexible control across different types of constraints. This framework combines two key mechanisms: a confidence-weighted control for point-wise constraints and a classifier-based control for managing statistical properties such as sums and averages over segments. Our methods achieve precise local control during the denoising inference stage while maintaining temporal coherence and integrating seamlessly, with any conditionally trained diffusion-based time series models. Extensive experiments across diverse datasets and models demonstrate its effectiveness. Our work bridges the gap between pure generative modeling and real-world time series editing needs, offering a flexible solution for human-in-the-loop time series generation and editing. The demonstration³ is provided for experiencing Time series editing.

1 Introduction

Time series data is highly prevalent in our daily lives. From financial markets to healthcare systems, time series generation (TSG) is increasingly vital for analysis and prediction [35]. Current approaches excel in unconditionally generating time series data, such as VAEs [17, 5, 29], GANs [11, 37, 22], and diffusion models [16, 27, 18, 2]. However, this unconditionally generated time series may be impractical in many applications. For example, in the retail industry, companies hope to forecast the revenue of an unreleased product. Unconditionally generated data may fail to satisfy desired properties that seem obvious to producers, i.e. sales should peak during holiday seasons or after coupons are mailed. Therefore, we hope to guide the generation process with certain prior knowledge. A simple case of prior knowledge would be data points that the generated curve must pass through.

*Equal contribution

†Corresponding author: shiweitong@tencent.com

³Time Series Editor is available here: Huggingface Space

An intuitive method would be to manually replace the point on the generated curve. However, this method does not consider the possibility that the point could be correlated with the rest of the time series curve. Therefore, this post-editing method risks generating out-of-distribution time series and does not leverage any information provided to us from partially observed data points or known trends. Thus, we hope to develop a principled way for generating time series while obeying certain user-specified rules, i.e., achieving controlled TSG.

We developed COCKTAILEDIT, a unifying framework named for its ability to mix and blend different types of constraints, to create precisely tailored time series outputs. Our framework embodies a wide range of various constraint types illustrated in Figure 1. We categorize these constraints by their range of focus. For example, we may have user-provided point-wise control, such as a single measurement on a fixed date with its confidence interval. Furthermore, we can also focus on global properties such as overall trend, periodicity, or specific data statistics (average, sum, optimizers, etc.). We regard these problems using an umbrella term called the Time Series Editing (TSE) challenge [15]; tackling this challenge means producing time series data that can simultaneously satisfy user-defined constraints while maintaining temporal coherence and distributional fidelity.

Methods for tackling strict subsets of the TSE challenge exist. For instance, control using provided trends in data can be achieved using models GANs [39, 4], VAEs [17, 29], Diffusion Models [3, 32, 38, 6, 33, 28, 1] and editing skills in Table 1. Diffusion models have also been applied to achieving point-wise control [7].

Table 1: Previous works for Time Series Editing. The RCGAN* and TimeGAN* adopt extra controls.

Features	GAN Models			Diffusion Models			
	RCGAN* [9]	TimeGAN* [39]	WGAN-COP [7]	DiffTime [7]	TimeWeaver [23]	TEdit [15]	CocktailEdit (Our)
Fixed Point	✓	✓	✓	✓	✗	✗	✓
Soft Point	✗	✗	✗	✗	✗	✗	✓
Trend	✓	✓	✓	✓	✓	✓	✓
Statistics	✓	✓	✓	✓	✗	✗	✓
Train with Condition	✓	✓	✓	✓	✓	✓	✗

However, none of these methods demonstrates satisfactory performance while being sufficiently fast and easy to use. They often incorporate prior knowledge by means of altering the model or training data, limiting flexibility for real-time human-centered modifications. Some methods choose to set a hard constraint on the learned distribution by embedding signals into the model architecture, e.g. time signals are injected into attention layers through additional embeddings, which limit the scope of this method and fail to consider possibilities of different kinds of human feedback/expected control signal [38, 6, 33, 28, 41, 1]. In light of these considerations, we propose a novel method that overcomes these difficulties and can be used off-the-shelf with no additional training process. Our method utilizes diffusion models because they offer a mathematically rigorous framework that surpasses GANs in perceptual quality while avoiding adversarial training difficulties [31, 30].

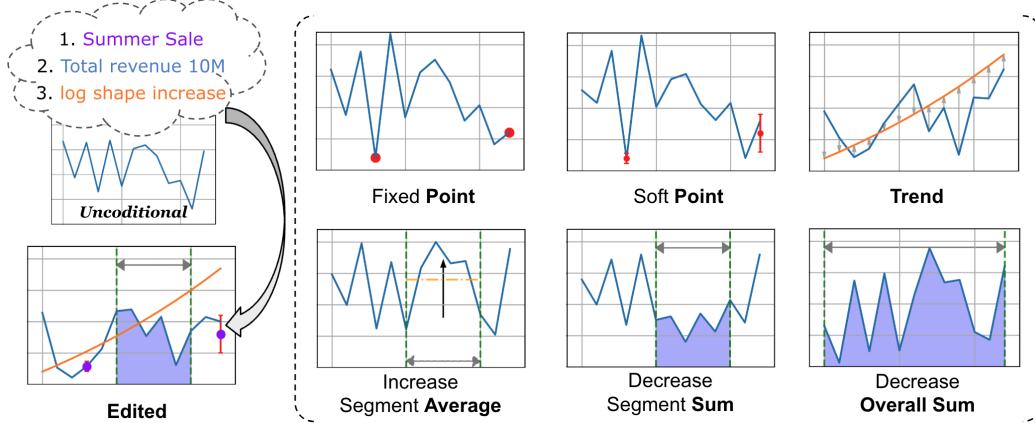


Figure 1: COCKTAILEDIT for Time Series Editing (TSE): From unconditional generation to fine-grained controlled time series generation, including (a) *point-level control* using fixed points and soft points with uncertainty, (b) *segment-level control* including trend and statistics metrics.

Through extensive experiments, we demonstrate that our method achieves precise control while maintaining temporal coherence and distributional fidelity. By bridging the gap between pure generative modelling and practical time series editing needs, our work offers a flexible solution for human-in-the-loop applications like revenue forecasting with expert knowledge integration and scenario analysis.

Our key contributions include:

- We propose the first unified framework for interactive TSE. Our algorithm combines multi-faceted constraints across three axes: time (point-wise and segment-wise constraints), type (exact value, trend, or more complicated functional statistics), and confidence level.
- We conduct extensive experiments and demonstrate the effectiveness of our methods across four diverse datasets while incorporating standard and novel experimental metrics suited towards our task.
- Our framework incorporates the constraints and operates only during the sampling process.

2 Preliminaries

2.1 Time Series Editing (TSE)

TSE is an umbrella term embodying difficulties from traditional time series tasks: prediction, imputation, and generation. However, while these classical tasks operate under specific, disjoint conditions – prediction uses historical data up to time t , imputation relies on partially observed segments, and unconditional generation requires no prior data – TSE must handle hybrid scenarios with multiple constraints.

To manage this complexity, TSE challenges are often decomposed along three dimensions: [1] *temporal granularity* (point-wise, segment-wise, and whole-series operations), [2] *conditioning type* (observed values, user-defined constraints, and learned patterns), and [3] *transformation scope* (\mathcal{X} for temporal range and \mathcal{Y} for value constraints). This decomposition allows for systematic editing across scales while maintaining consistency and supporting extensions from local to global modifications.

2.2 Problem Formulation

The objective of TSE is to generate time series data that replicates the statistical properties of real-world sequences while adhering to specific user-defined controls. Given a set of time series $S = \{\mathbf{x}_i\}_{i=1}^N$, where each $\mathbf{x}_i \in \mathbb{R}^{L \times D}$ (L is the sequence length, D is the number of features), we aim to develop a model f_θ that learned from S and parameterized by θ such that: $\mathbf{x} = f_\theta(\mathbf{C})$, \mathbf{C} represents the prior conditions. The model should be able to generate time series \mathbf{x} that satisfy both the learned distribution and the provided constraints.

2.3 Diffusion Models: DDPM and Conditional

Diffusion models, particularly Denoising Diffusion Probabilistic Models (DDPMs) [13, 26, 11], offer a robust backbone for addressing these TSE challenges. They tackle the controlled generation of time series, even with complex temporal dependencies and multi-scale patterns, by iteratively denoising data. This approach avoids common pitfalls such as the training instabilities of GANs or the reconstruction limitations of VAEs. The iterative nature of DDPMs, combined with guidance mechanisms like classifier-free and classifier-based methods [31], facilitates precise control at each step. This ensures distributional consistency and the preservation of essential local and global time series properties, effectively supporting editing constraints across diverse temporal granularities.

DDPM approximate a data distribution $q(\mathbf{x})$ by gradually adding noise to data samples and then learning to reverse this noising process. The forward process is a Markov chain:

$$q(\mathbf{x}_{1:K} \mid \mathbf{x}_0) = \prod_{k=1}^K q(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \quad (1)$$

$$q(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = \mathcal{N}\left(\mathbf{x}_k; \sqrt{1 - \beta_k} \mathbf{x}_{k-1}, \beta_k \mathbf{I}\right), \quad (2)$$

where $\beta_k \in [0, 1]$ controls the noise variance at each step. The reverse process is modeled by a parameterized Gaussian [13, 21]:

$$p_\theta(\mathbf{x}_{k-1} \mid \mathbf{x}_k) = \mathcal{N}\left(\mathbf{x}_{k-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_k, k), \sigma_k^2 \mathbf{I}\right). \quad (3)$$

To generate samples conditioned on extra inputs \mathbf{C} (e.g., labels or prompts), Denoising Diffusion models augment the reverse process:

$$p_\theta(\mathbf{x} \mid \mathbf{C}) = \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{C}), \text{ where} \quad (4)$$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{C}) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t, \mathbf{C}), \Sigma_\theta(\mathbf{x}_t, t)). \quad (5)$$

3 COCKTAILEDIT

To tackle the multi-grained TSE task, we propose COCKTAILEDIT that integrates point-wise and segment-wise controls, which includes two control mechanisms: 1) We view constraints as “masks” and use a teacher-forcing approach to guide the sampling process to enable confidence-weighted control. 2) Incorporate classifier-based control for managing segment-level aggregated statistical properties. These mechanisms are implemented during the reverse diffusion process and can be used simultaneously, enforcing constraints across different granularities. A complete table of notations is included in Appendix A.1.

3.1 Problem Setup

We begin by considering point-wise control as the fundamental problem. Given a time series \mathbf{x} with both observed values \mathbf{x}_{ob} at known indices $\Omega(\mathbf{x})$ and target values \mathbf{x}_{ta} which is the final target output, we aim to synthesize the target values using a diffusion model $p_\theta(\mathbf{x})$ trained on complete data while respecting the constraints. Formally, we define point-wise control constraints as: $\mathbf{C}_{\text{point}} = \{(\mathbf{t}_i, \mathbf{v}_i, \mathbf{c}_i, \mathbf{w}_i)\}_{i=1}^N$, where $\mathbf{t}_i \in \{1, \dots, L\}$ specifies the time index, $\mathbf{v}_i \in \mathbb{R}$ defines the expected value, $\mathbf{c}_i \in \{1, \dots, D\}$ means feature index, and $\mathbf{w}_i \in \mathbb{R}$ represents the confidence level.

3.2 Point-Wise Control

Replace-Based Masking In Diffusion-TS and Diffwave [40, 16], conditional sampling is achieved by replace-based value-infilling during the sampling process of the diffusion model. That is, at the given time indices in the condition, the intermediate noisy samples are replaced with observed values. However, since the observed values belong to the data distribution rather than the intermediate noisy distributions, given the fixed point values \mathbf{x}_0^{ob} , we inject the same level of noise to it as in the forward process.

$$q(\mathbf{x}_t^{\text{ob}} \mid \mathbf{x}_0^{\text{ob}}) = \mathcal{N}\left(\mathbf{x}_t^{\text{ob}}; \sqrt{\bar{\alpha}_t} \mathbf{x}_0^{\text{ob}}, (1 - \bar{\alpha}_t) \mathbf{I}\right), \quad (6)$$

$$\mathbf{x}_t^{\text{ta}} = \mathbf{x}_{t+1}^{\text{ta}} - \sqrt{\beta_t} \nabla_{\mathbf{x}_{t+1}^{\text{ta}}} \log p_\theta(\mathbf{x}_{t+1}^{\text{ta}} \mid \mathbf{x}_{t+1}^{\text{ta}}, \mathbf{x}_t^{\text{ob}}), \quad (7)$$

where $\bar{\alpha}_t = \prod_{i=1}^t (1 - \beta_i)$.

After obtaining \mathbf{x}_{ob}^t , we can then combine it with the intermediate noisy samples, similar to a teacher-forcing approach using discrete masks $\mathbf{m} \in \{0, 1\}^L$ for the indices relative to the given conditions.

$$\mathbf{x}_t = \mathbf{m} \odot \mathbf{x}_t^{\text{ob}} + (1 - \mathbf{m}) \odot \mathbf{x}_t^{\text{ta}}. \quad (8)$$

Confidence-Based Masking A natural extension is to consider continuous masks $\mathbf{m} \in [0, 1]^{L \times D}$ for enabling multivariate constraints and better generalization instead of $\mathbf{m} \in \{0, 1\}^L$. The continuous mask allows for interpolating between observed data and generated samples, which enables viewing masks as confidence weighting. We can thus interpret $\mathbf{C}_{\text{point}}$ as the following: $\mathbf{x}_i, \mathbf{y}_i, \mathbf{c}_i$ are grouped as hard constraints \mathbf{x}_0^{ob} , and the combined \mathbf{w}_i is the float mask \mathbf{m} , which is restricted to $\mathbf{w}_i \in [0, 1]$ and $\mathbf{m} \in [0, 1]^{L \times D}$. When $\mathbf{w}_i = 1$, we rely entirely on the observed data; when $\mathbf{w}_i = 0$, the model-generated sample is used. Appendix A.2 provides the formal proof.

Time-Dependent Weighted Guidance To emphasize stronger control in the later stages of the denoising process (i.e., as $t \rightarrow 0$), we introduce the time-dependent weight:

$$\omega_t = \exp(-\gamma \frac{t}{\text{num_timesteps}}), \quad (9)$$

and rewrite the linear combination between constraints and target values as

$$\mathbf{x}_t \leftarrow \omega_t \mathbf{m} \odot \mathbf{x}_t^{\text{ob}} + (1 - \omega_t \mathbf{m}) \odot \mathbf{x}_t^{\text{ta}}. \quad (10)$$

The visualized plot shows that the rate against the timestamp is in Appendix A.3.

Dynamic Error Adjustment on Mask Instead of a static mask, \mathbf{m} can be dynamically adjusted during the reverse diffusion steps based on intermediate predictions or error metrics. This allows the model to adaptively allocate more attention to regions with higher uncertainty or discrepancy from desired constraints: $\mathbf{m}_{t-1} = \mathbf{m}_t + \Delta\mathbf{m}_t$, where $\Delta\mathbf{m}_t$ is a function of the current estimation error $\|\mathbf{x}_t^{\text{ta}} - \mathbf{x}^{\text{target}}\|$ and with less inference steps.

3.3 Segment-wise Extension: Trend Control

Having a framework for confidence-weighted point-wise constraints allows natural extension to trend constraints, e.g. we may hope to generate a time-series that follows a reasonable periodic trend. The idea would be to compute \mathbf{x}_{ob}^t using the given trend, which are represented as time-dependent functions. Let \mathbf{L} encode the relationship between time and the expected function value over any segment. If \mathbf{L} is continuous, we can interpolate it to a discrete reference trend $\mathbf{l}_t \in \mathbb{R}$ for each time and corresponding values:

$$\mathbf{l}_t = \mathbf{L}_{t_s} + \frac{t - t_s}{t_e - t_s} (\mathbf{L}_{t_e} - \mathbf{L}_{t_s}), \quad t \in \{t_s, \dots, t_e\}. \quad (11)$$

By having multiple groups of $(\mathbf{t}_i, \mathbf{v}_i, \mathbf{c}_{\text{trend}}, \mathbf{w}_i)$ for $\mathbf{C}_{\text{point}}$ with $\mathbf{t}_i \in \{t_s, \dots, t_e\}$, $\mathbf{v}_i = \mathbf{l}_t$, we thus achieve trend control. Similarly, \mathbf{m}_i can be adjusted according to the user’s confidence level.

3.4 Combining Point-wise and Segment-wise Constraints

To handle different temporal scale constraints on $\mathbf{C}_{\text{point}}$, we aim to work with multiple masks at varying granularities (e.g., local, segment, global). In order to control the diffusion process at different temporal resolutions, we obtain the final masks \mathbf{m} using a reweighting scheme.

$$\mathbf{m} = \frac{\lambda_1 \mathbf{m}_{\text{local}} + \lambda_2 \mathbf{m}_{\text{segment}} + \lambda_3 \mathbf{m}_{\text{global}}}{\lambda_1 + \lambda_2 + \lambda_3}, \quad (12)$$

where λ_i are weighting coefficients ensuring that the combined mask maintains values within $[0, 1]$.

3.5 Statistics Control

We extend our notation set to formally define statistics control, which can be viewed as a variant of segment-wise control: $\mathbf{C}_{\text{segment}} = \{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{c}_j, \alpha_j, \mathbf{w}_j)\}_{j=1}^M$, where $\mathbf{s}_j, \mathbf{e}_j \in \{1, 2, \dots, L\}$ are the start and end indices of the segment, $\alpha_j \in \mathbb{R}$ the parameters for aggregated functions, and \mathbf{w}_j the confidence levels.

Additional Loss Term Following controllable diffusion models and [7], we inject a penalty term $\mathcal{L}_{\text{pen}}(\mathbf{x}_t)$ inspired by classifier guidance to enforce alignment with the given condition \mathbf{C} :

$$\mathcal{L}_{\text{pen}} = -\beta \log p_{\phi}(\mathbf{C} \mid \mathbf{x}_t), \quad (13)$$

with a hyperparameter β . Then, minimizing \mathcal{L}_{pen} effectively adds $-\beta \nabla_{\mathbf{x}_t} \log p_{\phi}(\mathbf{C} \mid \mathbf{x}_t)$ into the gradient flow [15].

Example of aggregated function – Sum

$$\mathcal{L}_{\text{sum}}[\mathbf{s}_j : \mathbf{e}_j] = \left(\sum_{i=s_j}^{e_j} \mathbf{x}_{t,i} - S_{\text{target}[\mathbf{s}_j : \mathbf{e}_j]} \right)^2, \quad (14)$$

where s_j and e_j denote the start and end indices of the segment. The final loss term is:

$$\mathcal{L}_{\text{sum}} = \omega_t \sum \beta_{\text{sum}[\mathbf{s}_j : \mathbf{e}_j]} \mathcal{L}_{\text{sum}[\mathbf{s}_j : \mathbf{e}_j]}, \quad (15)$$

where $\beta_{\text{sum}[\mathbf{s}_j : \mathbf{e}_j]}$ is the weight for the sum control term, and ω_t is the timestep-dependent weight that yields stronger guidance in the later stages of the denoising process defined in Eq. 9.

3.6 Combined Multi-Grained Control

Algorithm 1 in appendix presents our complete denoising control framework. Throughout the denoising process, we interleave point-wise floating mask control with segment-wise statistical control to gradually guide the denoising trajectory, requiring no model retraining or fine-tuning.

4 Experiments

4.1 Datasets

Our evaluation employs four diverse datasets across real-world and simulated scenarios. Table 2 summarizes the dataset specifications. The datasets include three real-world sources: ETTh⁴ for electricity transformer

⁴<https://github.com/zhouhaoyi/ETDataset>

measurements, fMRI⁵ for blood-oxygen-level-dependent time series and the Revenue dataset. The private Revenue dataset is sourced internally, containing revenue and two other features of hundreds of released video games. The simulated Sines dataset [39] is generated with varying frequencies, amplitudes, and phases.

Table 2: Dataset specifications summarizing the data sources, sequence lengths, and dimensionality of four datasets used.

Dataset	Type	Source	Length	Features
ETTh	Real-world	Electricity transformer	24	28
fMRI	Real-world	BOLD time series	24	50
Revenue	Real-world	Game sales	365	3
Sines	Simulated	Synthetic waves	24	5

The only difference is the fMRI dataset, which has a sequence length of 48 compared to the original with 24 timesteps.

4.2 Experimental Setup

Following the infilling training setup of the Diffusion-TS framework [40], we adopt the CSDI [32] and Diffusion-TS [40] as baseline fundamental models. The training stage uses the entire dataset in a multivariate time series aligned with the community. For all experiments, control signals were applied exclusively to the first channel ($c = 0$) to observe inter-channel influences while enabling multi-channel extensibility. All figures and results are based on the first channel unless stated otherwise. Training parameters and inference settings are expressed in Appendix B, such as choosing controlling points and confidence scores.

4.3 Evaluation Metrics

We employ multiple metrics across two key aspects: control accuracy and distribution quality. For *control accuracy*, we measure: (i) **Mean Absolute Difference (MAD)** between generated and target values at controlling points, defined as $MAD = \frac{1}{N} \frac{1}{A} \sum_{i=1}^N \sum_{j=1}^A |x_{\text{cond gen}(i,j)} - x_{\text{target}(i,j)}|$; and (ii) **Statistic Control Result** to evaluate adherence to target statistic constraints. For example, the control target function is the sum of series, we directly observe the actual sum value change to validate the controllability. For *distribution alignment* check, followed by many works [16, 40, 25, 28], we utilize: (i) **Discriminative & Predictive scores** to assess how well-generated sequences replicate real data patterns, where discriminative score is $|\text{accuracy} - 0.5|$ and predictive score uses mean absolute error (MAE) between predictions and ground truth; (ii) **Context-FID score** which leverages a pre-trained TS2Vec model to measure distribution-level alignment, with lower scores indicating better similarity; and (iii) **Correlational score** that quantifies feature-level covariance preservation. We also complement these quantitative metrics with **Kernel Density Function** visualizations to provide qualitative insights into distribution alignment.

5 Results

5.1 Point-Wise Control

We conducted experiments to examine the relationship between point confidence values and prediction accuracy across different datasets. These experiments aim to validate our hypothesis that higher confidence values lead to more precise point control and verify the behavior at extreme confidence values (0.01 and 1.0).

Table 3 with the Diffusion-TS backbone demonstrates that increasing confidence values leads to a consistent MAD reduction across all datasets, ultimately converging to 0.0 at maximum confidence. This convergence at confidence = 1.0 validates our theoretical guarantee that maximum confidence forces the point to be a fixed point. The monotonic and linear decrease of MAD with increasing confidence is visualized in Figure 2.

The second finding reveals that extreme target values (0.1 or 1.0) produce larger MAD values than moderate confidence levels (e.g., 0.5) across different datasets. For instance, on the fMRI Dataset, the MAD at $a_{\text{target}} = 0.1$ is 0.113, while at $a_{\text{target}} = 0.8$, it reduces to 0.052. This pattern, visible in Table 3, supports our hypothesis about the trade-off between very low and very high confidence values in controlling point selection. Comparing the MAD of the Original Dataset and Unconditional across the three target values, the point approach demonstrates a smaller MAD even with a confidence score of 0.01. This indicates that point-wise control helps generate time series data closer to the target value at a specified time. Furthermore, the pattern of largest and

⁵<https://www.fmrib.ox.ac.uk/datasets/netsim>

Table 3: Point Control MAD of the given point indices in different setups of different confidence levels and target values.

ETTh Dataset						
Target Value \ Confidence	Original	0.00 (Uncon)	0.01	0.50	1.00	Average
0.1	.0661	.0614	.0233	.0038	.0000	.0090
0.8	.7201	.7363	.0734	.0400	.0000	.0378
0.8	.5775	.6021	.1293	.0622	.0000	.0638
1.0	.7775	.8021	.2219	.1007	.0000	.1075
Average	.4988	.5102	.1274	.0594	.0000	
fMRI Dataset						
Target Value \ Confidence	Original	0.00 (Uncon)	0.01	0.50	1.00	Average
0.1	.4423	.4409	.0288	.0050	.0000	.0113
0.8	.2620	.2632	.0129	.0027	.0000	.0052
1.0	.4577	.4591	.0163	.0027	.0000	.0063
Average	.3873	.3877	.0193	.0035	.0000	
Sine Dataset						
Target Value \ Confidence	Original	0.00 (Uncon)	0.01	0.50	1.00	Average
0.1	.6530	.6617	.2511	.1116	.0000	.1209
0.8	.1466	.1455	.0700	.0377	.0000	.0359
1.0	.2470	.2383	.1179	.0588	.0000	.0589
Average	.3489	.3485	.1463	.0694	.0000	

Table 4: Sum Control Average summation value for various weights of sum control. “Original”: the original training set; “Uncon”: Unconditional generated samples. The **biggest** and **smallest** MAD are labelled in each dataset. (Diffusion-TS)

Dataset	Original	Uncon	Target Value		
			-100.0	20.0	50.0
Sine	17.881	18.086	<u>7.114</u>	20.146	20.991
Revenue	80.194	76.619	<u>52.675</u>	55.380	58.585
ETTh	1.924	1.535	<u>0.802</u>	8.502	10.323
fMRI	12.990	12.980	<u>4.675</u>	17.192	19.811
Dataset	Original	Uncon	Weight Value		
			1	50	100
Sine	17.881	18.086	<u>7.110</u>	7.114	7.117
Revenue	80.194	76.619	<u>52.767</u>	52.675	52.725
ETTh	1.924	1.535	0.798	0.802	0.800
fMRI	12.990	12.980	4.678	<u>4.675</u>	4.682

smallest MAD values aligns with those in the original and unconditional versions, suggesting that point-wise control preserves the original distribution and MAD for each dataset while reducing its MAD magnitude.

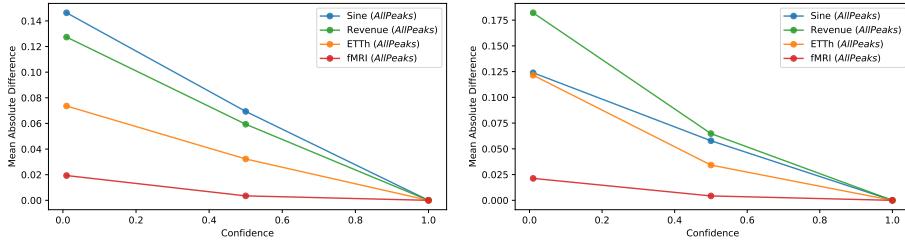


Figure 2: Point Control The influence of confidence across datasets. The right figure is examined when enabling the dynamic error adjustment and time-dependent weight, while the left figure is not. “AllPeaks” means the average of all experiments of point-wise control. More details are in Appendix C.4.

5.2 Segment-Wise Statistical Control

Our experiments investigate two key aspects of the aggregated statistic adjustment mechanism: (1) the effectiveness of different target values in steering the **segment sum** and **sequence sums**, and (2) the impact of **weight parameters** on control strength. These experiments aim to validate whether our method can reliably guide sequences toward desired sum targets while maintaining data fidelity.

Whole Sequence Sum As shown in Table 4, the controlled sequences consistently respond to different target values across all datasets. For the Revenue dataset, when targeting 150.0, the sequence sum increases significantly from 76.6 to 117.9, while targeting -100.0 reduces it to 52.7. Similar patterns are observed in other datasets - ETTh shows controlled variation from 1.5 to 11.1 (100) and 0.8 (-150), etc. Moreover, Revenue shows increasing steps: +2.705 from -100.0 to 20.0, +3.206 from 20.0 to 50.0, and +59.283 from 50.0 to 150.0. Both patterns demonstrate fine-grained control for smaller adjustments and the ability to make substantial changes when needed. Figures in Appendix D.1 visualize more controlled time series with different targets.

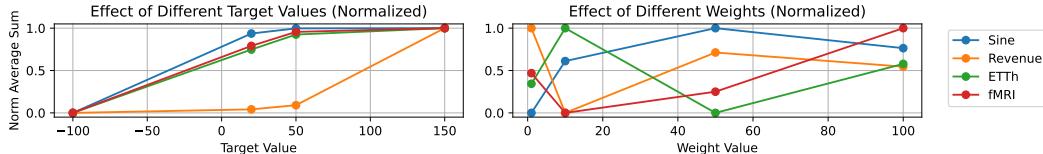


Figure 3: Normalized comparison of sum control effectiveness across datasets. Values are scaled relative to dataset-specific ranges to enable direct comparison between different domains. Lines show progressive convergence toward target sums. (Diffusion-TS)

The weight parameter experiments reveal that varying weights from 1 to 100 produces only minor changes in the resulting sums. Target value rather than the control weight is the primary driver of loss-based control performance. Then, the weight parameters may be simplified in practical applications since they do not contribute substantially to control performance. The normalized trends in Figure 5.2 further support this observation.

Segment Sum To evaluate segment-wise control, we tested 3 segments: $(0.2L, 0.4L)$, $(0.4L, 0.6L)$, $(0.6L, 0.8L)$ with target value 150. Figure 4 shows that each controlled segment demonstrates increased

sum adjustments. The effectiveness is particularly visible in the ETTh and Revenue datasets. The controlled segments exhibit clear increases in area under the curve when targeting higher sums with Diffusion-TS, while the effect of CSDI controlling is not obvious.

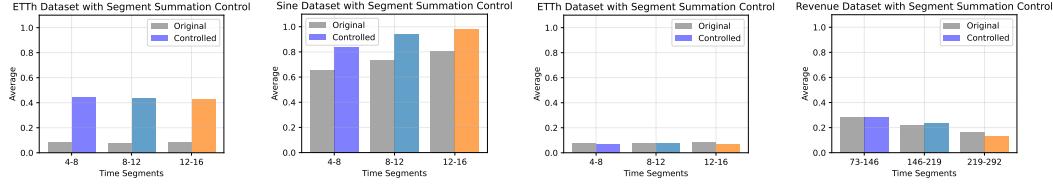


Figure 4: Segment-wise sum control results for different datasets. The shaded area represents the controlled segment, with the corresponding average sum value across different time segments. The datasets from left to right are: ETTh, Revenue, fMRI and Sine.

5.3 Distribution Alignment

Table 5 demonstrates that Diffusion-TS achieves better distribution matching than CSDI in unconditional generation across all datasets, with scores approaching 0, particularly for ETTh (0.034) and Sines (0.019). However, both Point-wise and Statistics controls lead to increased divergence, suggesting a trade-off between control and distribution preservation. The comprehensive table can be found in Appendix C.3.

Table 5: Discriminative score for our method on CSDI and Diffusion-TS (DTS). The lower the score, the more similar the distribution of generated time series with original datasets.

Model	Control	ETTh	Revenue	fMRI	Sines
Our - CSDI	Unconditional	0.361±0.007	0.245±0.164	0.306±0.021	0.017±0.007
	Point-wise	0.470±0.003	0.313±0.046	0.482±0.004	0.430±0.038
	Statistics	0.373±0.007	0.272±0.055	0.377±0.019	0.034±0.007
Our - DTS	Unconditional	0.034±0.026	0.209±0.185	0.089±0.033	0.019±0.008
	Point-wise	0.437±0.004	0.393±0.030	0.495±0.001	0.460±0.011
	Statistics	0.477±0.003	0.426±0.032	0.498±0.001	0.451±0.029

Our time efficiency analysis from Table 6 reveals CSDI’s computational advantage, processing samples 3-9x faster than Diffusion-TS across all datasets. The runtime scales primarily with sequence length, as evidenced by the Revenue dataset’s higher processing times. Notably, both point-wise and statistics-based controls add minimal computational overhead, maintaining consistent performance across control configurations. Feature dimensionality has a secondary but observable impact on processing time.

Table 6: Time efficiency analysis of our method compared on CSDI and Diffusion-TS (DTS). The results show the average time per sample for each dataset and control configuration. 500 examples per batch with an average of 5 batches.

Method	Name	Dataset		Time Per Sample (ms)		
		Seq	Features	Uncon	Point-Wise	Statistics
Our - CSDI	Energy	24	28	0.064±0.000	0.064±0.004	0.064±0.001
	fMRI	24	50	0.116±0.000	0.115±0.001	0.117±0.001
	Revenue	365	3	1.000±0.001	1.016±0.022	1.001±0.013
Our - DTS	Sines	24	5	0.018±0.000	0.017±0.001	0.018±0.001
	Energy	24	28	0.278±0.017	0.304±0.002	0.292±0.001
	fMRI	24	50	0.376±0.154	0.232±0.883	0.549±0.274
	Revenue	365	3	9.163±0.411	9.580±5.126	9.223±0.044
	Sines	24	5	0.046±0.015	0.059±0.020	0.048±0.001

6 Related Work

Traditional generative tasks employed GANs [37, 24, 39, 4], VAEs [8, 17], Diffusion Models [3, 32, 38, 6, 33, 28, 1, 15, 42], and Flow Matching [14], establishing foundational techniques for synthetic time series generation.

6.1 Time Series Generation Models

TimeGAN [39] introduced temporal-aware adversarial training, while subsequent works like C-TimeGAN [4] and CGAN-TS [22] incorporated conditional generation capabilities. Recent Transformer-based architectures

[29, 36] have further enhanced representation capabilities for temporal patterns. Early control mechanisms focused on global attributes through models like TimeVAE [8], TimeGAN [39], and C-TimeGAN [4]. Recent advances include hierarchical approaches [34] and attention-based mechanisms [20]. Notably, CGAN-TS [22] and ControlTS [35] introduced attribute-based and temporal feature control.

6.2 Diffusion Models for Time Series

The adaptation of diffusion models to time series data has seen rapid advancement through several key developments. Early works like TimeGrad [27] and CSDI [32] established the viability of diffusion models for temporal data, particularly in handling missing value imputation and uncertainty quantification. These foundations led to architectural innovations including TimeDiT [6], which introduced specialized temporal attention mechanisms, and Diff-MTS [28], which enhanced multivariate time series generation through improved cross-channel modelling.

Recent advances have focused on both architectural improvements and control mechanisms. The Latent Diffusion Transformer [10] demonstrated efficient generation through compressed latent spaces, while RATD [19] introduced robust attention mechanisms for handling temporal dependencies. Score-CDM [41] and DiffusionBridge [25] advanced controlled generation through score-based approaches and bridge construction methods respectively. These developments have enabled successful applications across diverse domains, from healthcare monitoring [1] and industrial systems [33] to financial forecasting [38, 12].

While existing methods show promise, they primarily focus on global trend control and distribution matching, lacking, fine-grained control at the individual timestamp level except DiffTime and Guided-DiffTime [7], working on the point-wise constrained time series generation. However, their points control needs additional training.

7 Conclusion

We present the COCKTAILEDIT for generalized Time Series Editing that enables fine-grained control and global statistical manipulation without model retraining. It is achieved via float masking and score-based guidance for statistical properties. The interleaving nature of the two mechanisms allows them to be seamlessly combined, enabling sophisticated editing operations that respect both fine-grained constraints and coarse-grained properties. This makes our framework particularly suitable for real-world applications where domain experts need to incorporate specific knowledge while maintaining overall statistical validity. Figure G demonstrates the practical interactive editing interface to enable the intuitive manipulation of temporal data like Photoshop. The control mechanism is not perfect and the obvious distribution drift still exists. Our method can be further improved by incorporating more advanced control mechanisms and carefully considering the trade-off between control precision and distribution preservation.

References

- [1] Edmonmd Adib, Amanda S. Fernandez, Fatemeh Afghah, and John J. Prevost. Synthetic ecg signal generation using probabilistic diffusion models. *IEEE Access*, 11:75818–75828, 2024.
- [2] Juan Miguel Lopez Alcaraz and Nils Strothoff. Diffusion-based time series imputation and forecasting with structured state space models. *Version published by Transactions on Machine Learning Research in 2022 (TMLR ISSN 2835-8856) https://openreview.net/forum?id=hHilbk7ApW*, August 2022.
- [3] Yihao Ang, Qiang Huang, Yifan Bao, Anthony K. H. Tung, and Zhiyong Huang. Tsgbench: Time series generation benchmark. September 2023.
- [4] Gaby Baasch, Guillaume Rousseau, and Ralph Evins. A conditional generative adversarial network for energy use in multiple buildings using scarce data. *Energy and AI*, 5:100087, 2021.
- [5] Yifan Bao, Yihao Ang, Qiang Huang, Anthony K. H. Tung, and Zhiyong Huang. Towards controllable time series generation. March 2024.
- [6] Defu Cao, Wen Ye, Yizhou Zhang, and Yan Liu. Timedit: General-purpose diffusion transformers for time series foundation model. September 2024.
- [7] Andrea Coletta, Sriram Gopalakrishnan, Daniel Borrajo, and Svitlana Vytenko. On the constrained time-series generation problem. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 61048–61059. Curran Associates, Inc., 2023.
- [8] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. November 2021.

- [9] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. June 2017.
- [10] Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):11979–11987, Mar. 2024.
- [11] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.
- [12] Mohamed Hamdouche, Pierre Henry-Labordere, and Huyêñ Pham. Generative modeling for time series via schrödinger bridge. April 2023.
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. June 2020.
- [14] Yang Hu, Xiao Wang, Lirong Wu, Huatian Zhang, Stan Z. Li, Sheng Wang, and Tianlong Chen. Fm-ts: Flow matching for time series generation. November 2024.
- [15] Baoyu Jing, Shuqi Gu, Tianyu Chen, Zhiyu Yang, Dongsheng Li, Jingrui He, and Kan Ren. Towards editing time series. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [16] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.
- [17] Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. March 2023.
- [18] Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23009–23022. Curran Associates, Inc., 2022.
- [19] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. October 2024.
- [20] Xinhe Liu and Wenmin Wang. Deep time series forecasting models: A comprehensive survey. *Mathematics*, 12(10), 2024.
- [21] Caspar Meijer and Lydia Y. Chen. The rise of diffusion models in time-series forecasting. January 2024.
- [22] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8983–8991, 2021.
- [23] Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep Chinchali. Time weaver: A conditional time series generation model. March 2024.
- [24] Nam Nguyen and Brian Quanz. Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9117–9125, 2021.
- [25] Jinseong Park, Seungyun Lee, Woojin Jeong, Yujin Choi, and Jaewook Lee. Leveraging priors via diffusion bridge for time series generation. August 2024.
- [26] Lucas Pinheiro Cinelli, Matheus Araújo Marins, Eduardo Antônio Barros da Silva, and Sérgio Lima Netto. Variational autoencoder. In *Variational Methods for Machine Learning with Applications to Deep Networks*, pages 111–149. Springer, 2021.
- [27] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8857–8868. PMLR, 18–24 Jul 2021.
- [28] Lei Ren, Haiteng Wang, and Yuanjun Laili. Diff-mts: Temporal-augmented conditional diffusion-based aigc for industrial time series toward the large model era. *IEEE Transactions on Cybernetics*, 54(12):7187–7197, 2024.
- [29] Alexander Sommers, Logan Cummins, Sudip Mittal, Shahram Rahimi, Maria Seale, Joseph Jaboure, and Thomas Arnold. A survey of transformer enabled time series synthesis. June 2024.

- [30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. October 2020.
- [31] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. November 2020.
- [32] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24804–24816. Curran Associates, Inc., 2021.
- [33] Muhang Tian, Bernie Chen, Allan Guo, Shiyi Jiang, and Anru R Zhang. Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models. *Journal of the American Medical Informatics Association*, 31(11):2529–2539, 09 2024.
- [34] José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: A survey. *Big Data*, 9(1):3–21, February 2021.
- [35] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. July 2024.
- [36] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–12, 2023.
- [37] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, 2020.
- [38] Fangkai Yang, Wenjie Yin, Lu Wang, Tianci Li, Pu Zhao, Bo Liu, Paul Wang, Bo Qiao, Yudong Liu, Mårten Björkman, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Diffusion-based time series data imputation for cloud failure prediction at microsoft 365. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE ’23*, pages 2050–2055. ACM, November 2023.
- [39] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [40] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. March 2024.
- [41] S. Zhang, S. Wang, H. Miao, H. Chen, C. Fan, and J. Zhang. Score-cdm: Score-weighted convolutional diffusion model for multivariate time series imputation. May 2024.
- [42] Lingqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. September 2023.

A Methodology Additional Details

A.1 Summary of Variants

Table 7: Summary of variables and their Meanings

Symbol	Meaning
N	Number of time series
$\mathbf{x}_i \in \mathbb{R}^{L \times D}$	i -th time series, length L , dimension D
$S = \{\mathbf{x}_i\}_{i=1}^N$	Training dataset of time series
θ	Model parameters of f_θ
\mathbf{C}	Prior/conditioning input
T	Total forward/backward diffusion steps
K	Set of number of gradient steps per diffusion step
$\beta_k \in [0, 1]$	Noise variance schedule
\mathbf{x}_t	Noisy sample at diffusion step t
t_i	Time index in the sample \mathbf{x}
\mathbf{v}_i	Corresponding y value of the time index in sample \mathbf{x}
μ_θ, σ_k^2	Mean and variance in reverse diffusion
$\Omega(\mathbf{x}), \bar{\Omega}(\mathbf{x})$	Observed and missing indices
$\mathbf{x}_{ob}, \mathbf{x}_{ta}$	Observed and target parts of a time series
$\mathbf{m} \in [0, 1]^{L \times D}$	Confidence/float mask
ω_t	Time-dependent weight in reverse steps
\mathbf{C}_{point}	Point-wise control with confidence
$\mathbf{C}_{segment}$	Segment-wise control constraints
$\mathcal{L}_{sum}, \mathcal{L}_{pen}, \mathcal{L}_{statistics}$	Loss terms enforcing constraints
$\gamma, \eta, \beta_{sum[s_j:e_j]}$	Additional scale factors for guidance

A.2 Formal Proof of Confidence-Based Masking

Proof. Let $p(x)$ be the jointly trained diffusion model over both observed and missing dimensions, and let $p(x_{ta} | x_{ob})$ be the target conditional distribution. Denote $p_t(x_{ob}^t, x_{ta}^t)$ as the distribution of (x_{ob}^t, x_{ta}^t) at iteration t . We show that as $t \rightarrow 0$, $p_t(x_{ta}^t | x_{ob}^t)$ converges to $p(x_{ta} | x_{ob})$.

I. Forward-Process Marginals. By construction, x_{ob}^t at each step is drawn from the forward process $q(x_{ob}^t | x_{ob}^0)$, which is a Gaussian transition that preserves the exact marginal of the known dimensions x_{ob} . Formally, for each t ,

$$x_{ob}^t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_{ob}^0, (1 - \bar{\alpha}_t) \mathbf{I}).$$

Hence, $p_t(x_{ob}^t)$ remains consistent with the correct marginal distribution $\prod_{\Omega(x)} p(x_{ob})$.

II. Denoising of Missing Entries. The reverse step for missing entries x_{ta}^t is governed by

$$x_{ta}^t \leftarrow x_{ta}^{t+1} - \sqrt{\beta_t} \nabla_{x_{ta}^{t+1}} \log p_\theta(x_{ta}^{t+1} | x_{ta}^{t+2}, x_{ob}^t).$$

As shown in [31], iteratively applying the reverse diffusion steps in this score-based framework converges to sampling from $p(x_{ta} | x_{ob}^t)$.

III. Replace and Float-Mask Consistency. Whether we *replace* x_{ob}^t completely or *blend* it with a float mask \mathbf{m} :

$$x^t = \mathbf{m} \odot x_{ob}^t + (1 - \mathbf{m}) \odot x_{ta}^t,$$

the observed indices remain consistent with their forward-sampled values. This ensures that at every iteration, the joint distribution respects the known-data constraints. The 0.0 value in masks means no restriction on this point, and 1.0 means the fixed point.

IV. Convergence to the Conditional. Consider $p_t(x_{ta}^t | x_{ob}^t)$. By the score-based argument (the forward-reverse chain forming a time-indexed Markov process), the mixture of denoising steps and partial resets of observed entries yields

$$\lim_{t \rightarrow 0} p_t(x_{ta}^t | x_{ob}^t) = p(x_{ta} | x_{ob}),$$

where the convergence follows from the fact that each reverse diffusion step corrects the noise injected in the forward pass, conditioned on the known x_{ob}^t .

Thus, replace-based or float-mask imputation each preserves x_{ob} 's marginals and iteratively refine x_{ta} until the distribution of the missing entries matches $p(x_{ta} | x_{ob})$. \square

Implementation Details The float mask \mathbf{m} can be adjusted to control the influence of observed data on the imputed sequence. By iterative sampling from the diffusion model and applying the floating mask, we can generate time series data that respects observed values while conforming to the correct conditional distribution.

1. Initialize x^T from Gaussian noise.
2. For $t = T$ down to 1:
 - Forward-process sampling (for x_{ob}^t): $x_{ob}^t \sim q(x_{ob}^t | x_{ob}^0)$.
 - Denoise missing parts (for x_{ta}^t): perform gradient step using $\log p_\theta$.
 - Apply float mask (optional): fuse observed and imputed regions via \mathbf{m} .

A.3 Time-Dependent Weight

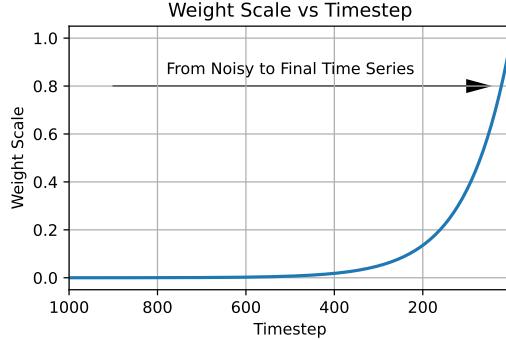


Figure 5: Time-dependent weight schedule during denoising process. The exponential decay ($\gamma = 5.0$) provides stronger control signals in later timesteps while allowing smoother adjustments near completion.

A.4 COCKTAILEDIT Algorithm

Algorithm 1 DDPM Denoising with COCKTAILEDIT

Require: Gradient scale η , trade-off coefficient γ , conditional data \mathbf{x}_a , time-dependent weight ω_t

```

1: Initialize  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T$  to 1 do
3:   // Step 1: Predict and Refine Sample
4:    $[\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_b] \leftarrow p_\theta(\mathbf{x}_t, t, \theta)$ 
5:    $\mathcal{L}_1 = \|\mathbf{x}_a - \hat{\mathbf{x}}_a\|_2^2$ 
6:    $\mathbf{x}_{t-1} \leftarrow \mathcal{N}(\mu(p_\theta(\mathbf{x}_t, t, \theta), \mathbf{x}_t), \Sigma)$ 
7:    $\mathcal{L}_2 = \|\mathbf{x}_{t-1} - \mu(p_\theta(\mathbf{x}_t, t, \theta), \mathbf{x}_t)\|_2^2 / \Sigma$ 
8:   // Step 2: Statistics Control
9:    $\mathcal{L}_{statistics} = \omega_t \sum \beta_{sum[s_j:e_j]} \mathcal{L}_{sum[s_j:e_j]}$ 
10:   $\tilde{\mathbf{x}}_0 = p_\theta(\mathbf{x}_t, t, \theta) + \eta \nabla_{\mathbf{x}_t} (\mathcal{L}_1 + \gamma \mathcal{L}_2 + \mathcal{L}_{statistics})$ 
11:   $\mathbf{x}_{t-1} \leftarrow \mathcal{N}(\mu(\tilde{\mathbf{x}}_0, \mathbf{x}_t), \Sigma)$ 
12:  // Step 3: Point-Wise Control
13:   $\mathbf{x}_{t-1} \leftarrow \omega_t \mathbf{m}^{ob} \odot \mathbf{x}_t^{ob} + (1 - \omega_t \mathbf{m}^{ob}) \odot \mathbf{x}_{t-1}$ 
14: end for

```

B Experiment Additional Details

Table 8: Training hyperparameters and settings for each dataset.

Method	Dataset			LR	Training Train Steps	Batch Szie	Inference DDPM Timesteps
	Name	Seq Length	Features				
Our - CSDI	Energy	24	28	1.00E-03	25000	64	200
	fMRI	24	50	1.00E-03	15000	64	200
	Revenue	365	3	2.00E-03	2230	64	200
	Sines	24	5	1.00E-03	12000	128	50
Our - Diffusion-TS	Energy	24	28	1.00E-05	25000	64	1000
	fMRI	24	50	1.00E-05	15000	64	1000
	Revenue	365	3	2.00E-05	2230	64	500
	Sines	24	5	1.00E-05	12000	128	500

We run all training and inferencing of all experiments with NVIDIA L40S GPUs. All experiments are fixed on the random seed with 2024. Table 8 summarizes the training hyperparameters for each dataset and method. We use the Adam optimizer with a learning rate of 1.00×10^{-3} for CSDI and 1.00×10^{-5} for Diffusion-TS. The training steps are set to 25,000 for CSDI and 15,000 for Diffusion-TS. The batch size is 64 for CSDI and 128 for Diffusion-TS. The number of diffusion steps is set according to the different datasets.

For point-wise control, we placed target points (called “anchor” in the following) at normalized values $\mathbf{v}_{\text{anchor}} \in \{0.1, 0.8, 1.0\}$ at relative temporal positions $t_{\text{anchor}} \in \{0.1L, 0.3L, 0.5L, 0.7L, 0.9L\}$. The confidence levels are set to $\mathbf{c}_{\text{anchor}} = \{0.01, 0.50, 1.0\}$ at the corresponding anchor indices, and the other positions were set to 0.0 during mask converting.

For the segment-wise control, we test on segments $\{(s, e)\} = \{(0.2L, 0.4L), (0.4L, 0.6L), (0.6L, 0.8L), (0, L)\}$, $(0, L)$ represents the whole sequence. We choose “sum” as the aggregated function, then $\alpha_{\text{sum}} \in \{-100, 20, 50, 100\}$ as the targeted aggregated statistics, weight $\beta_{\text{sum}} \in \{1, 10, 50, 100\}$. We note that baseline sums vary significantly between datasets due to sequence length and data scaling, we will include the various sequence length results, such $L \in \{96, 192, 384\}$ in the camera-ready version.

C Point-Wise Control

C.1 Demonstrates

The following figures demonstrate the effectiveness of point-wise control across different datasets varying this confidence and target value with Diffusion-TS and CSDI. We can see as the confidence increases, the model respects the anchor points more strictly.

C.1.1 Pure Float Mask Control (Diffusion-TS)

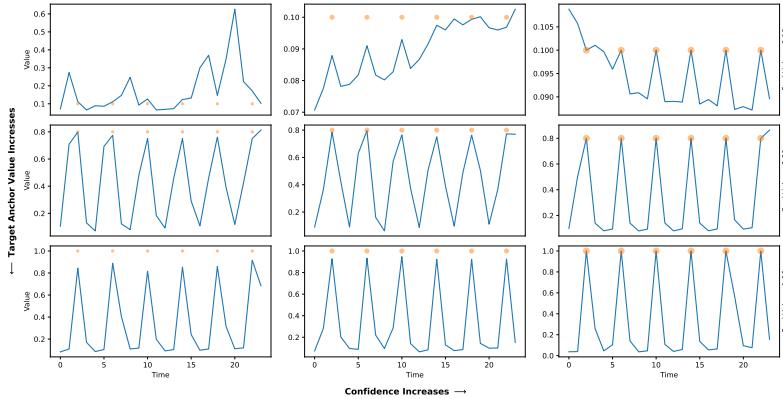


Figure 6: Demonstration of Point-Wise Control in ETTh datasets with multiple control points and confidences

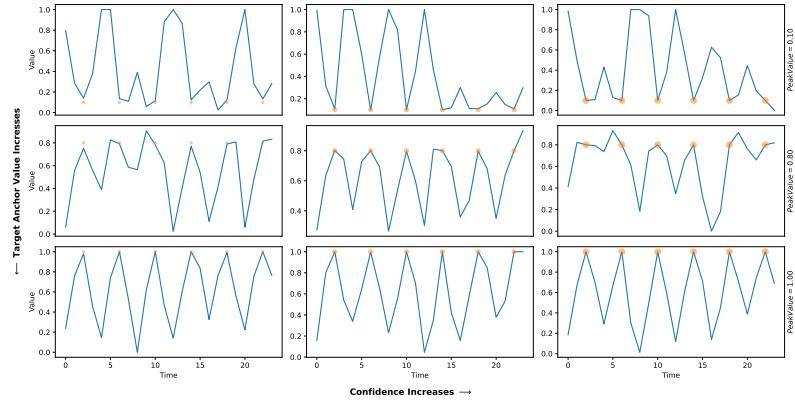


Figure 7: Demonstration of Point-Wise Control in fMRI datasets with multiple control points and confidences.

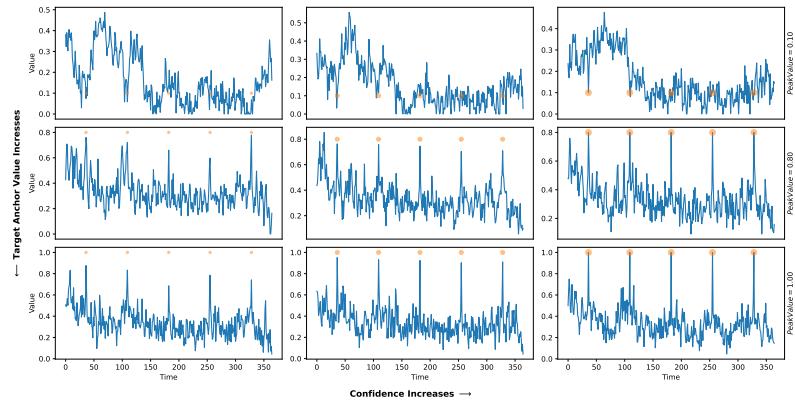


Figure 8: Demonstration of Point-Wise Control in Revenue datasets with multiple control points and confidences

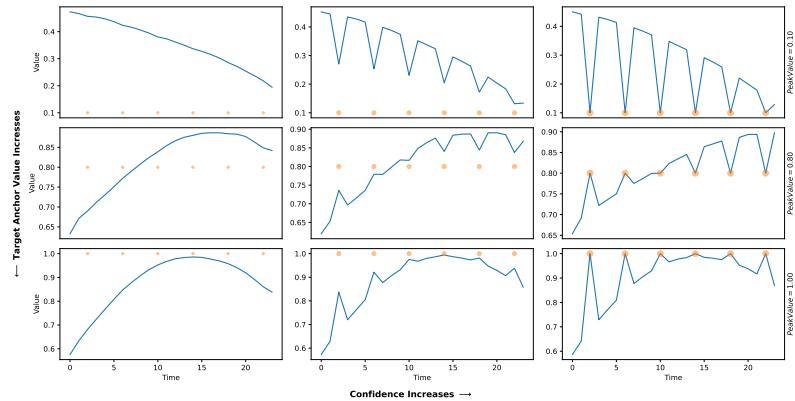


Figure 9: Demonstration of Point-Wise Control in Synthetic sine wave datasets with multiple control points and confidences

C.1.2 Float Mask Control with Extensions (Diffusion-TS)

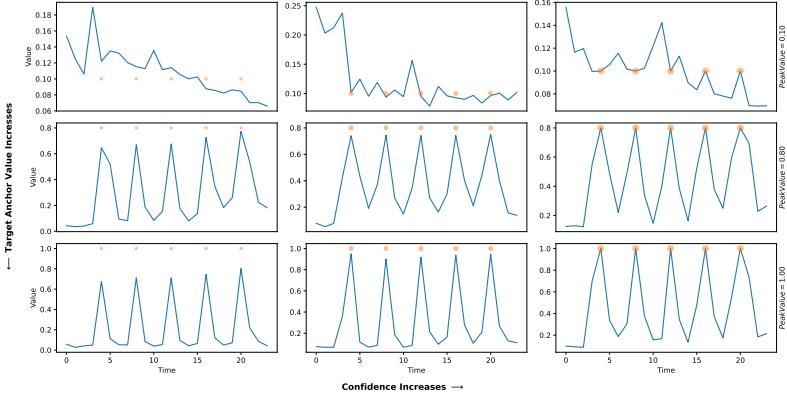


Figure 10: Demonstration of Point-Wise Control in ETTh datasets with multiple control points and confidences

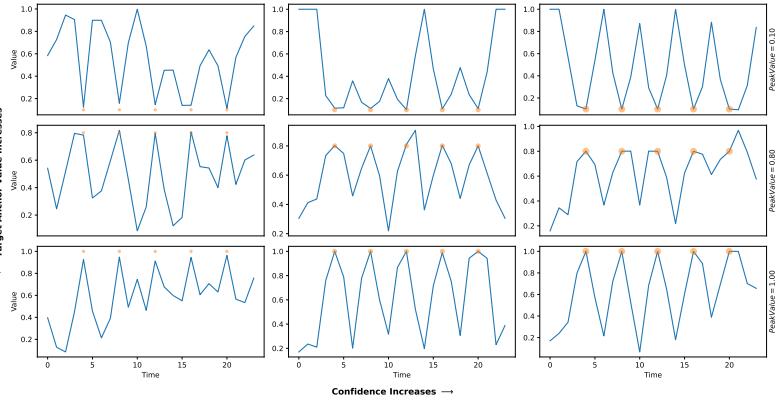


Figure 11: Demonstration of Point-Wise Control in fMRI datasets with multiple control points and confidences.

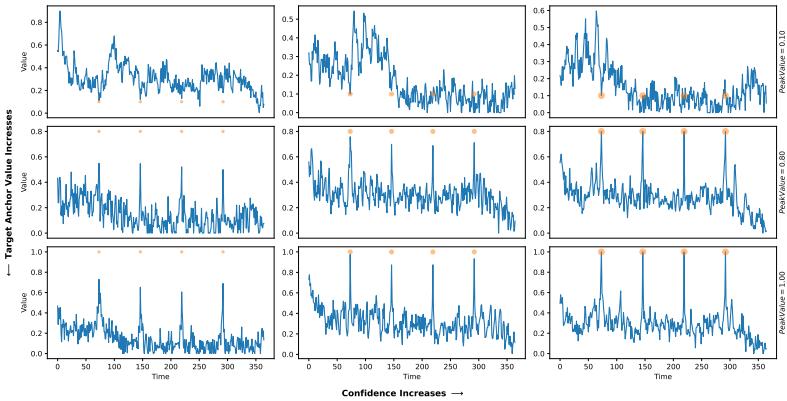


Figure 12: Demonstration of Point-Wise Control in Revenue datasets with multiple control points and confidences

C.1.3 Float Mask Control with Extensions (CSDI)

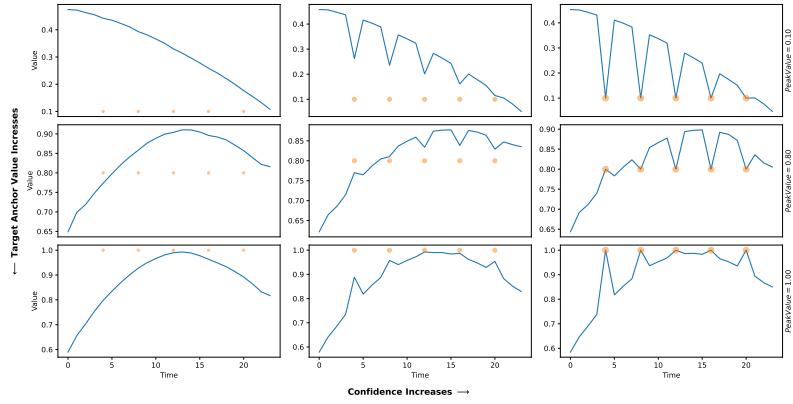


Figure 13: Demonstration of Point-Wise Control in Synthetic sine wave datasets with multiple control points and confidences

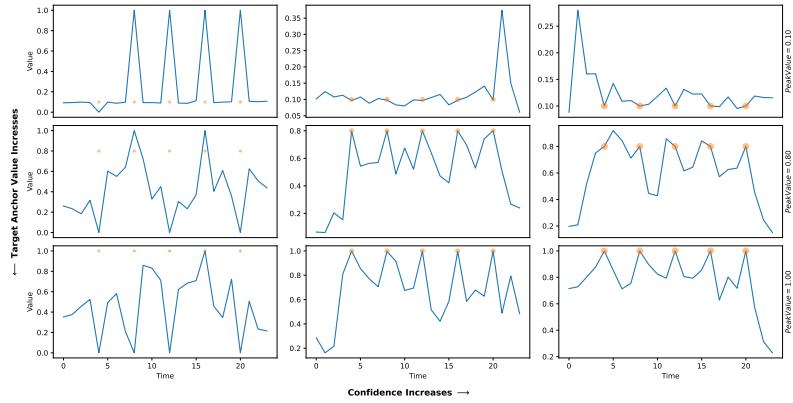


Figure 14: Demonstration of Point-Wise Control in ETTh datasets with multiple control points and confidences

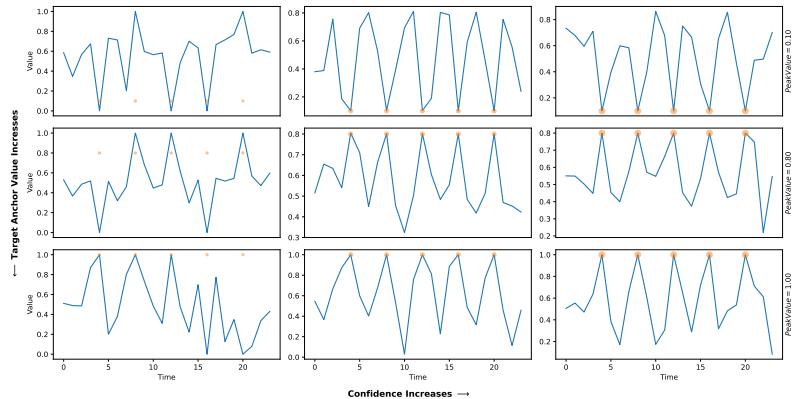


Figure 15: Demonstration of Point-Wise Control in fMRI datasets with multiple control points and confidences.

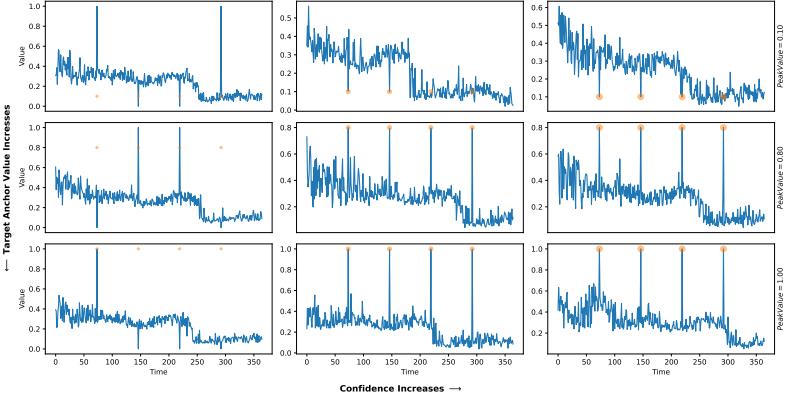


Figure 16: Demonstration of Point-Wise Control in Revenue datasets with multiple control points and confidences

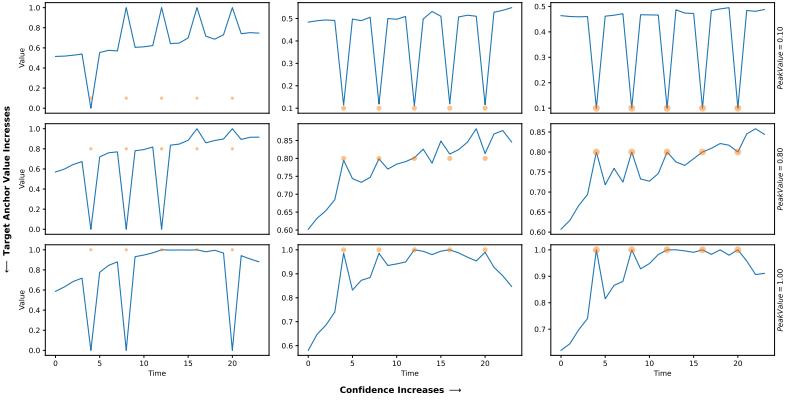


Figure 17: Demonstration of Point-Wise Control in Synthetic sine wave datasets with multiple control points and confidences

C.2 KED of Point-Wise Control

The following plots of Kernel Density Estimation (KDE) clearly demonstrate how the distribution peaks (purple dash line) shift towards the anchor points as confidence increases. For instance, in the ETTh dataset, the most pronounced shift occurs at anchor points under the highest confidence level. The model generates sequences that accurately respect anchor points while preserving the dataset's inherent distributional characteristics. In Figure C.2.1's middle row, where the target value is 0.8, increasing confidence levels cause the peaks (y-value density) of controlled results (purple line) to intensify and converge toward the target value. This pattern is consistently observed across all datasets.

C.2.1 Pure Float Mask Control (Diffusion-TS)

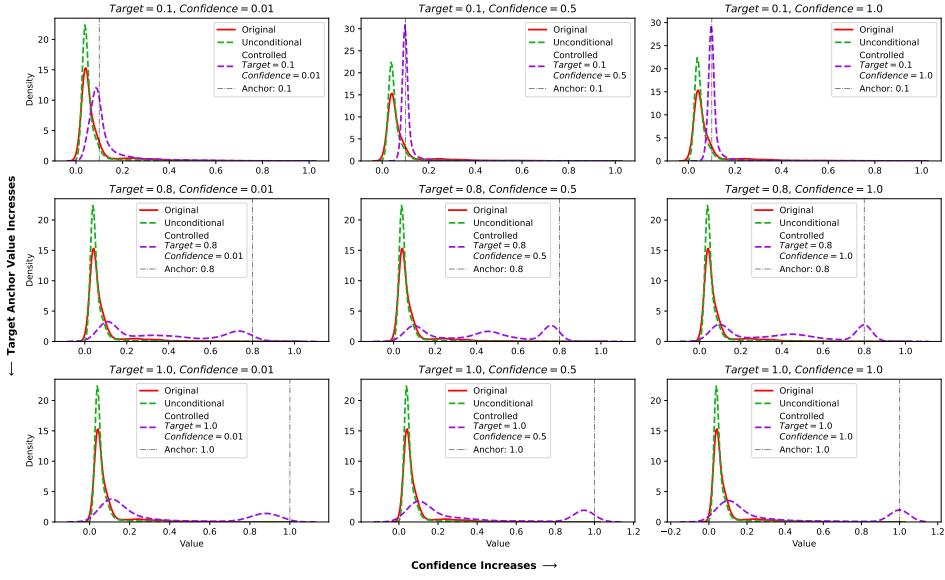


Figure 18: KDE analysis of ETTh dataset generation.

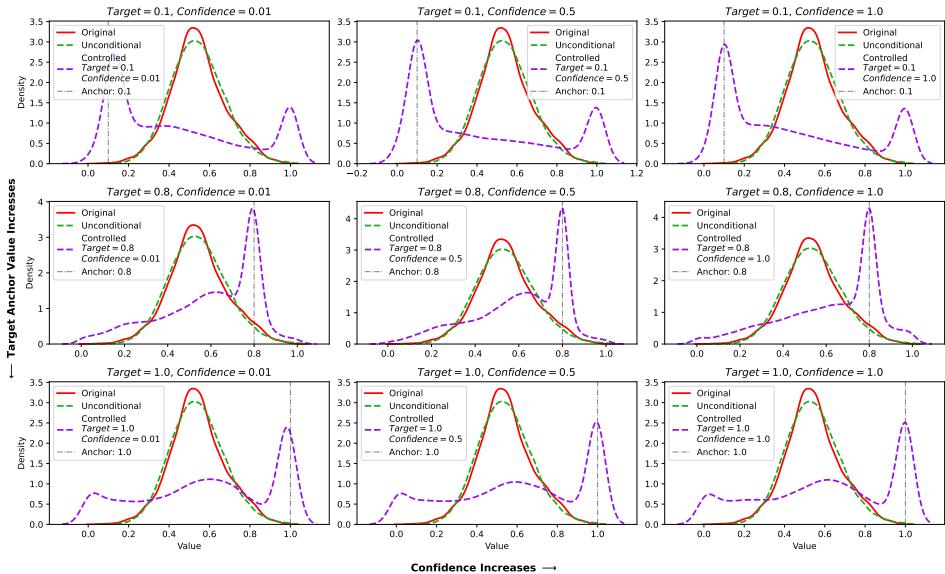


Figure 19: KDE analysis of fMRI dataset generation.

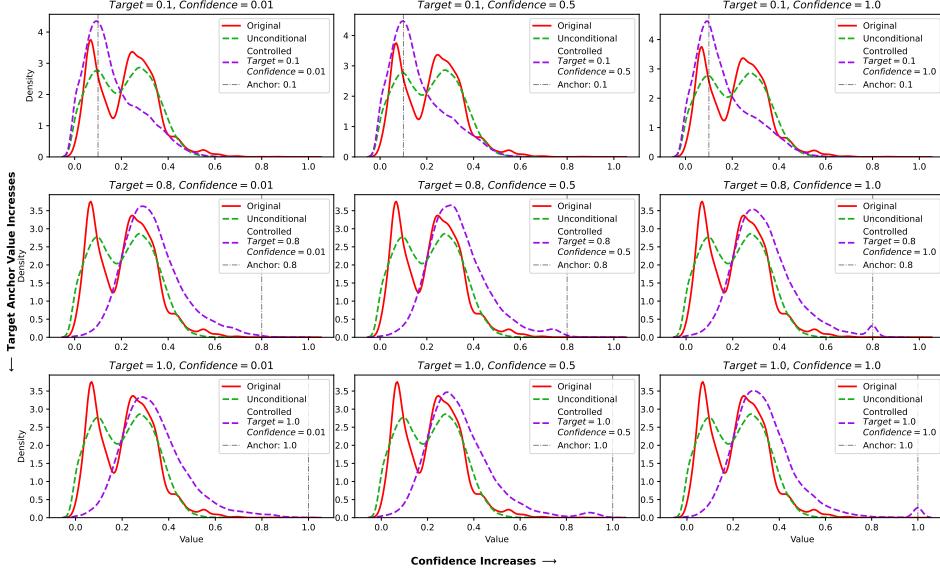


Figure 20: KDE analysis of Revenue dataset generation.

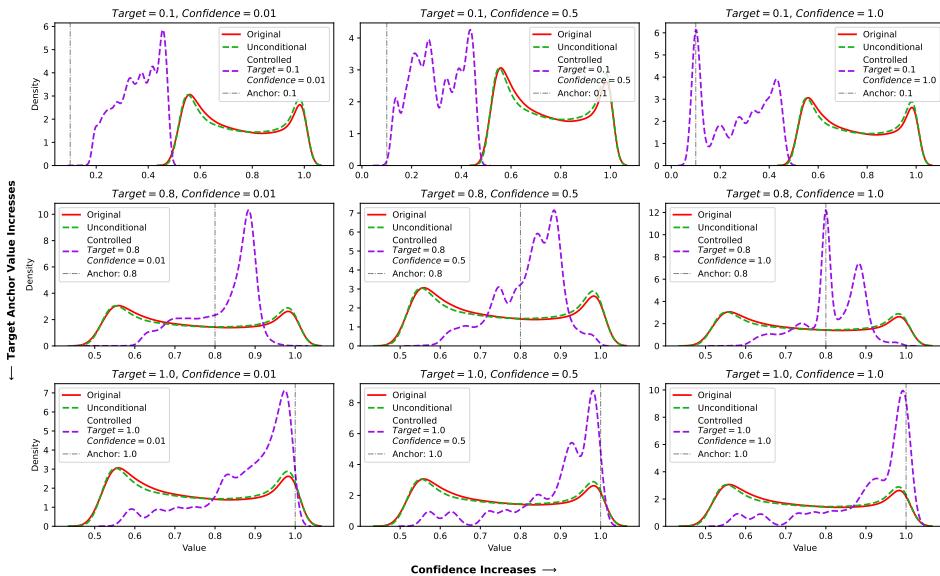


Figure 21: KDE analysis of synthetic sine wave dataset generation.

C.2.2 Float Mask Control with Extensions (Diffusion-TS)

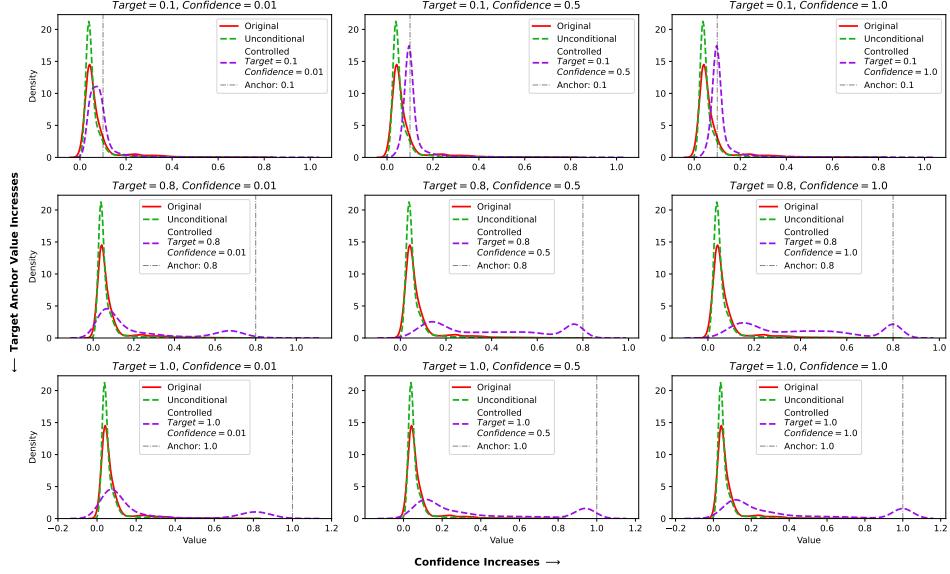


Figure 22: KDE analysis of ETTh dataset generation.

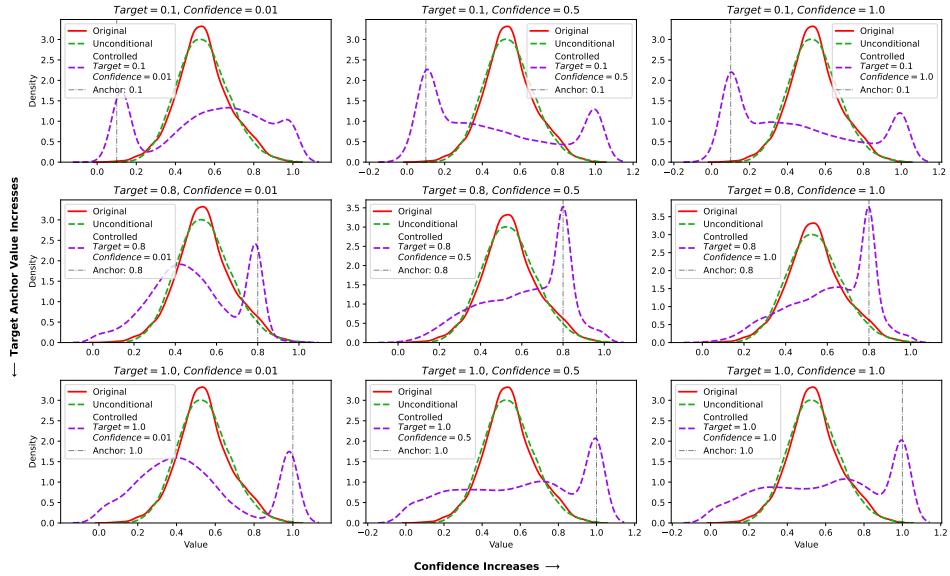


Figure 23: KDE analysis of fMRI dataset generation.

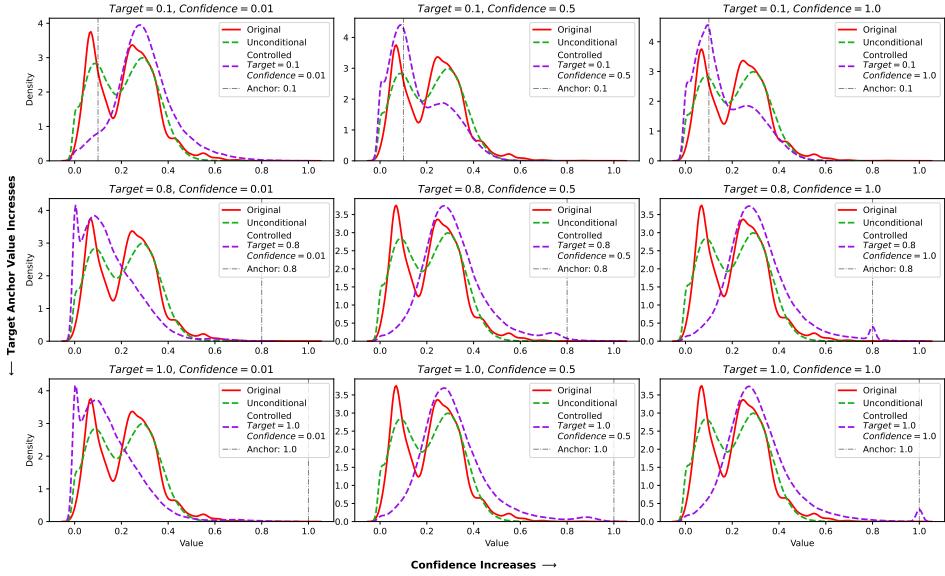


Figure 24: KDE analysis of Revenue dataset generation.

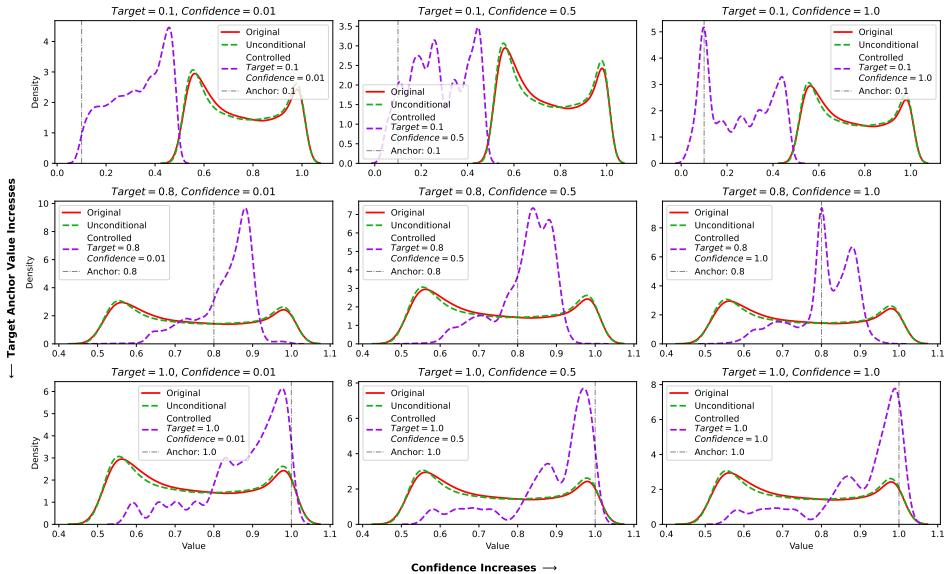


Figure 25: KDE analysis of synthetic sine wave dataset generation.

C.2.3 Float Mask Control with Extensions (CSDI)

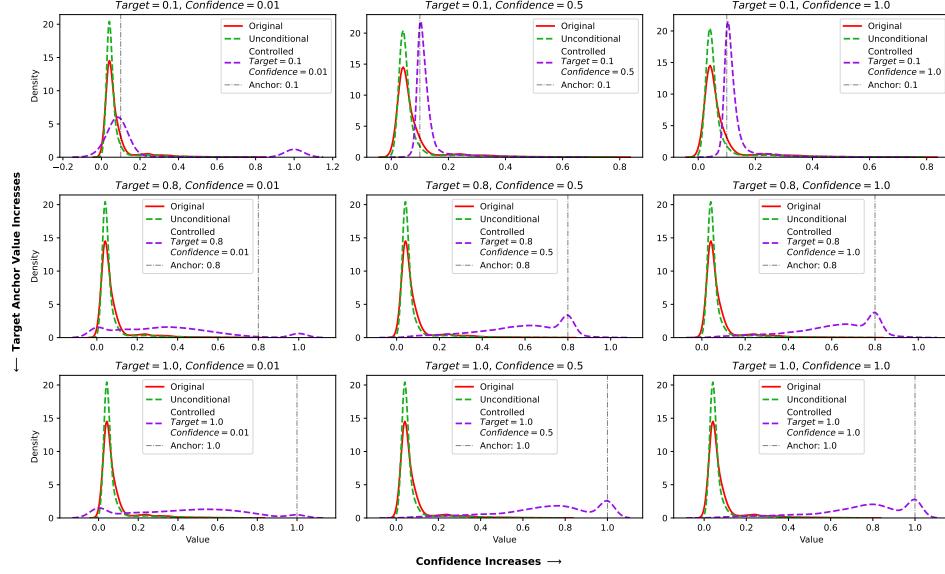


Figure 26: KDE analysis of ETTh dataset generation.

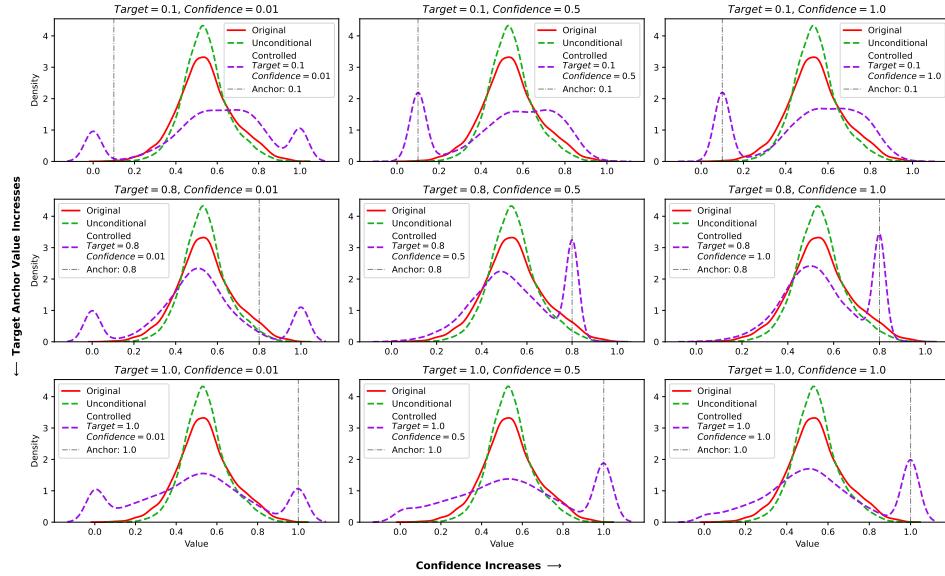


Figure 27: KDE analysis of fMRI dataset generation.

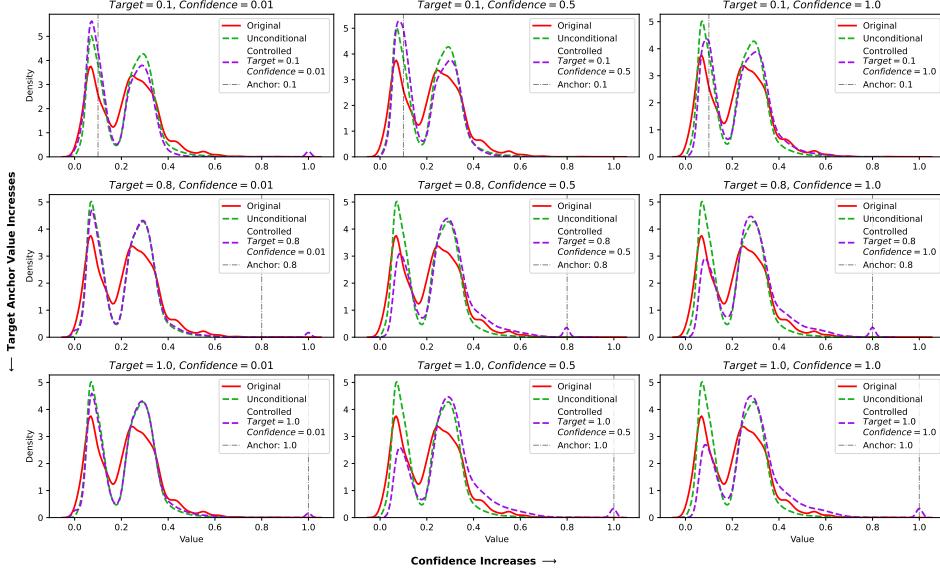


Figure 28: KDE analysis of Revenue dataset generation.

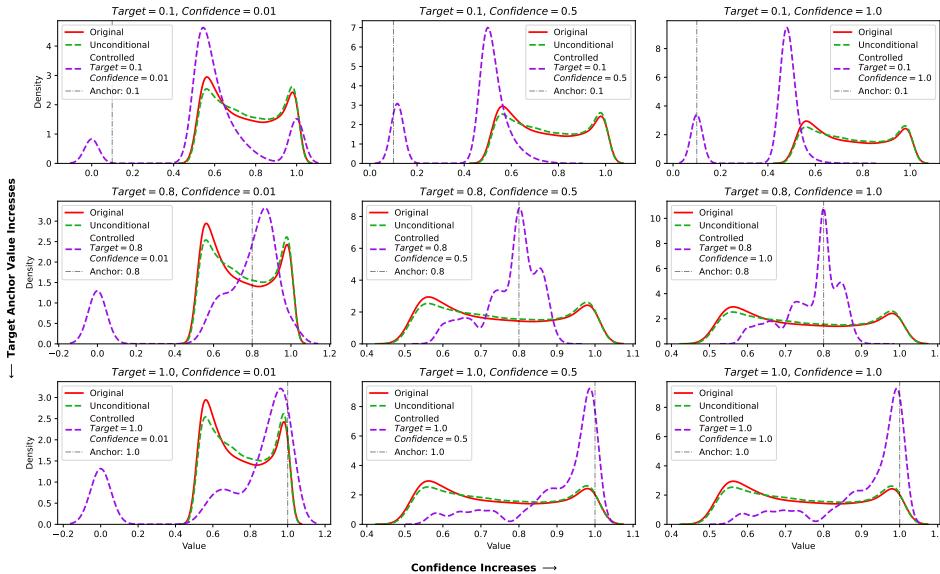


Figure 29: KDE analysis of synthetic sine wave dataset generation.

C.3 Supplement Metrics

The Discriminative, Predictive, Context-FID, and Correlational scores help quantify distribution shifts under point-wise control. Table C.3 demonstrates that all metrics increase significantly after applying control, indicating that the control signals effectively influence the generated time series. But our method makes them more distinguishable from the original distribution, which is the improved direction for following research.

Table 9: Supplemental metrics for point-wise control performance across different datasets and target values. The results show discriminative, predictive, context-FID, and correlational scores for each dataset and control configuration. Lower scores indicate better performance.

Metrics	Control Signal	Dataset			
		ETTh	Revenue	fMRI	Sine
Discriminative Score (Lower is Better)	Unconditional	0.103±0.042	0.082±0.093	0.141±0.037	0.031±0.023
	Confidence=0.01	0.497±0.005	0.382±0.267	0.500±0.000	0.457±0.027
	Confidence=0.5	0.496±0.006	0.282±0.323	0.499±0.001	0.494±0.004
	Confidence=1.0	0.498±0.001	0.009±0.025	0.498±0.000	0.374±0.264
Predictive Score (Lower is Better)	Unconditional	0.256±0.002	0.065±0.026	0.103±0.002	0.094±0.000
	Confidence=0.01	0.302±0.008	0.181±0.002	0.134±0.010	0.120±0.007
	Confidence=0.5	0.305±0.014	0.179±0.005	0.141±0.010	0.118±0.012
	Confidence=1.0	0.335±0.019	0.175±0.009	0.139±0.009	0.116±0.008
Context-FID Score (Lower is Better)	Unconditional	0.108±0.007	1.230±0.284	0.260±0.024	0.034±0.005
	Confidence=0.01	7.797±0.644	4.654±1.389	15.671±2.920	4.169±1.139
	Confidence=0.5	6.973±1.514	5.921±0.711	15.463±1.888	10.842±2.103
	Confidence=1.0	7.856±1.326	7.083±0.512	13.854±0.830	8.906±1.168
Correlational Score (Lower is Better)	Unconditional	2.313±0.743	0.038±0.013	2.672±0.091	0.066±0.009
	Confidence=0.01	9.321±0.764	0.122±0.007	16.699±0.453	0.255±0.013
	Confidence=0.5	8.445±0.675	0.119±0.006	15.488±0.174	0.468±0.016
	Confidence=1.0	9.640±0.835	0.121±0.011	17.259±0.434	0.345±0.034

C.4 Point-Wise Control Analysis

Here, we demonstrate again the complete aggregated Mean Absolute Deviation (MAD) of point-wise control across all datasets, providing solid evidence for the effectiveness of point-wise control in maintaining anchor points.

C.4.1 Pure Float Mask Control

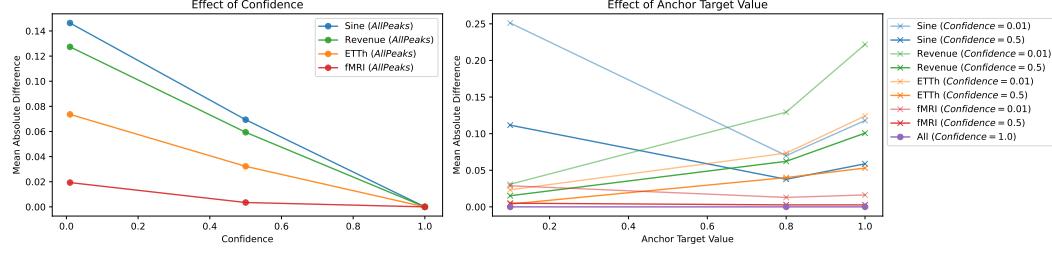


Figure 30: Different combinations of confidence levels and target values across all datasets. (Diffusion-TS)

C.4.2 Float Mask Control with Extensions

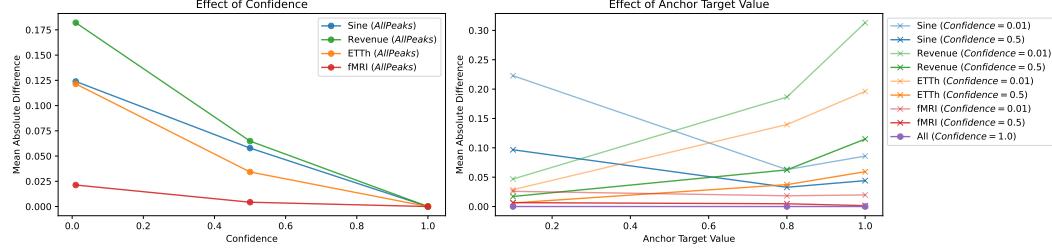


Figure 31: Different combinations of confidence levels and target values cross all datasets. (Diffusion-TS)

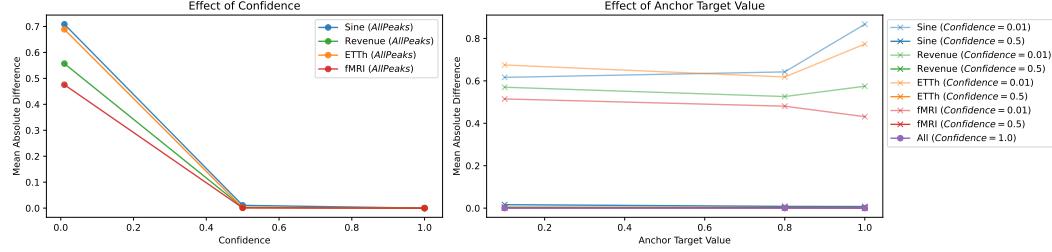


Figure 32: Different combinations of confidence levels and target values across all datasets. (CDSI)

D Statistic Control

This section provides supplementary materials for analyzing statistical control, focusing on both Sum Control and Segment-Wise Sum Control.

D.1 Demonstration

The following figures demonstrate the effectiveness of sum control across different datasets. As the target value increases (from left to right in each row), the model generates sequences that successfully adhere to the sum constraints while maintaining the dataset's inherent distributional properties. For segment-wise sum control, the results consistently show that the model tends to increase the overall sequence sum value, which aligns with the objective of preserving the original distribution learned from the dataset.

D.1.1 Whole Time Series Summation Control (Diffusion-TS)

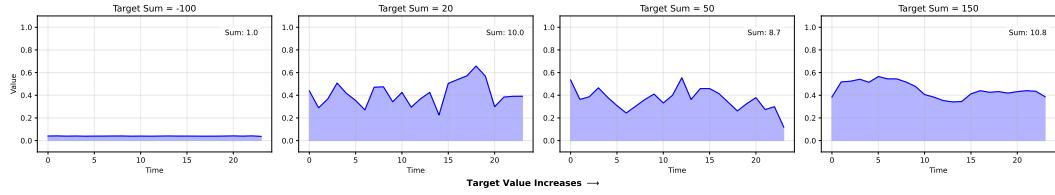


Figure 33: Demonstration of Sum Control on ETTh dataset.



Figure 34: Demonstration of Sum Control on fMRI dataset.

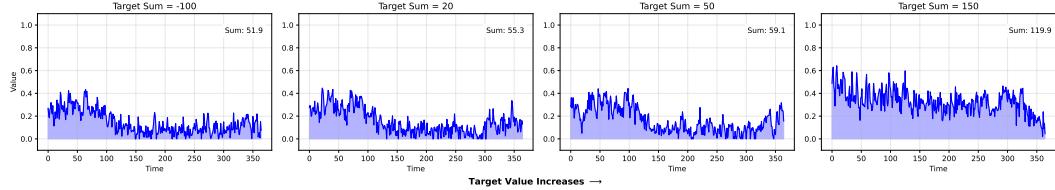


Figure 35: Demonstration of Sum Control on Revenue dataset.

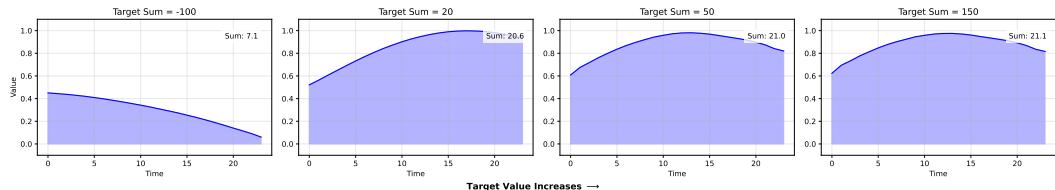


Figure 36: Demonstration of Sum Control on synthetic sine wave dataset.

D.1.2 Whole Time Series Summation Control (CSDI)

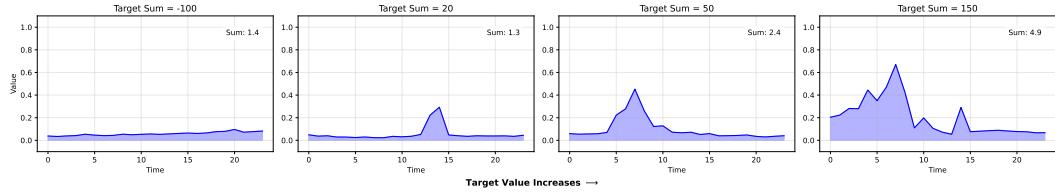


Figure 37: Demonstration of Sum Control on ETTh dataset.

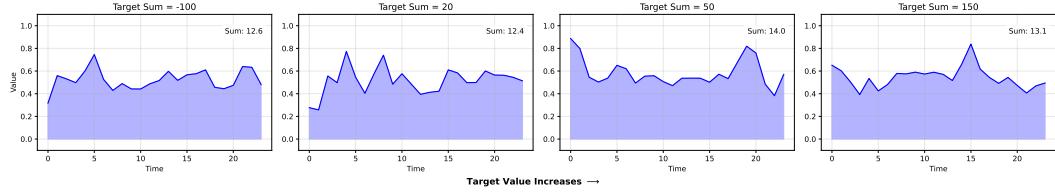


Figure 38: Demonstration of Sum Control on fMRI dataset.

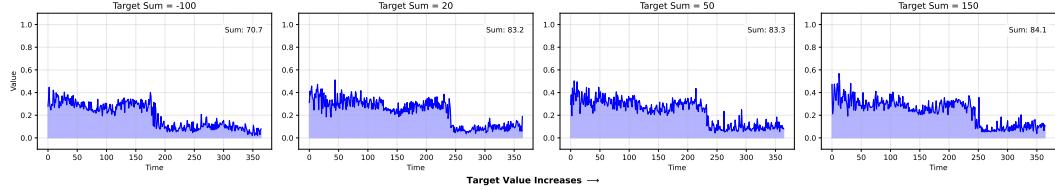


Figure 39: Demonstration of Sum Control on Revenue dataset.

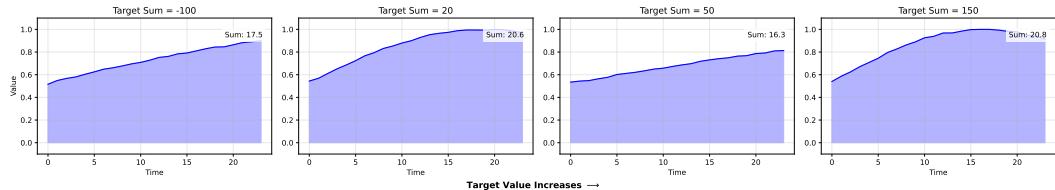


Figure 40: Demonstration of Sum Control on synthetic sine wave dataset.

D.1.3 Segment Wise Sum Control (Diffusion-TS)

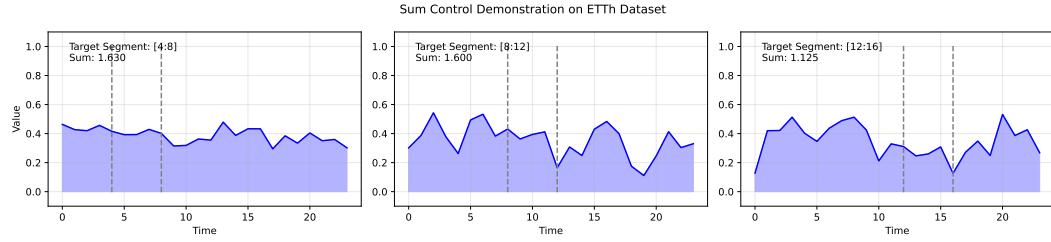


Figure 41: Demonstration of Segment-Wise Sum Control on ETTh dataset.

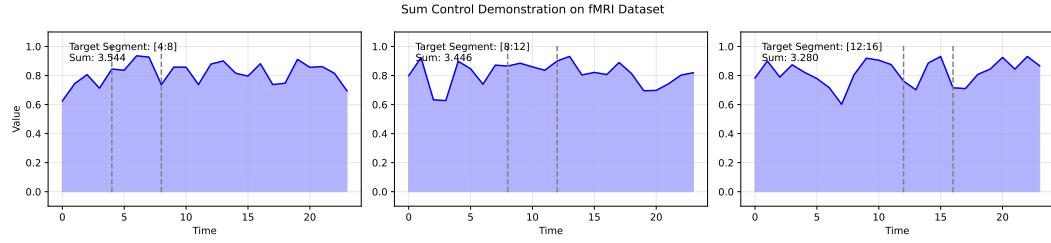


Figure 42: Demonstration of Segment-Wise Sum Control on fMRI dataset.

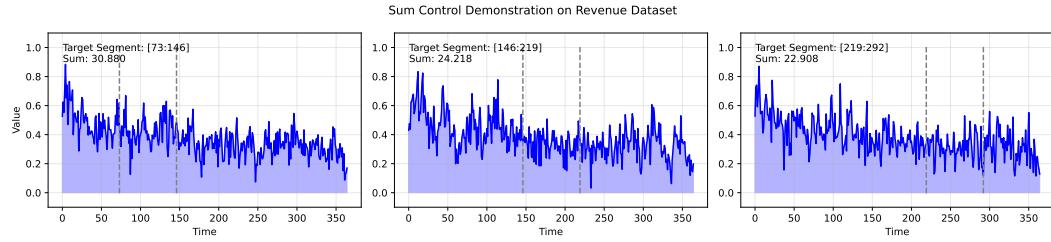


Figure 43: Demonstration of Segment-Wise Sum Control on Revenue dataset.

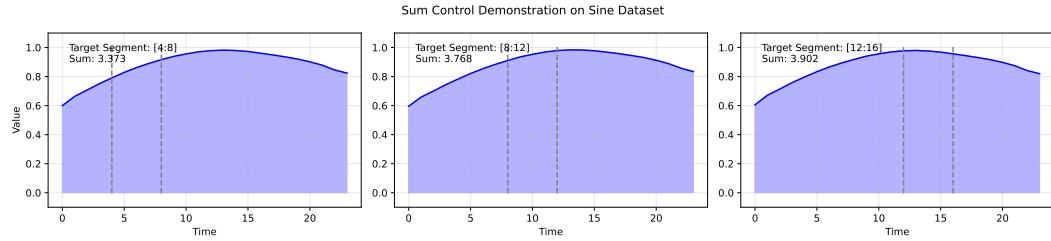


Figure 44: Demonstration of Segment-Wise Sum Control on synthetic sine wave dataset.

D.1.4 Segment Wise Sum Control (CSDI)

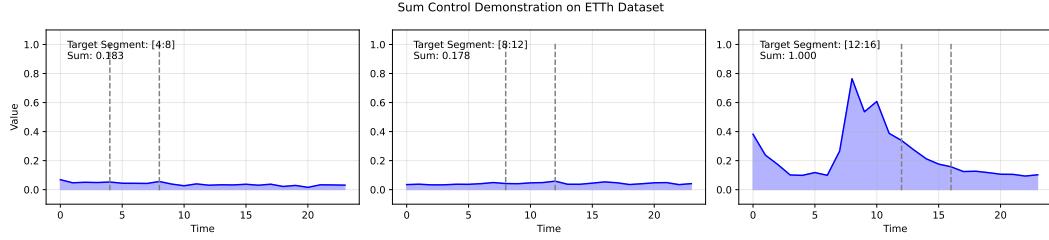


Figure 45: Demonstration of Segment-Wise Sum Control on ETTh dataset.

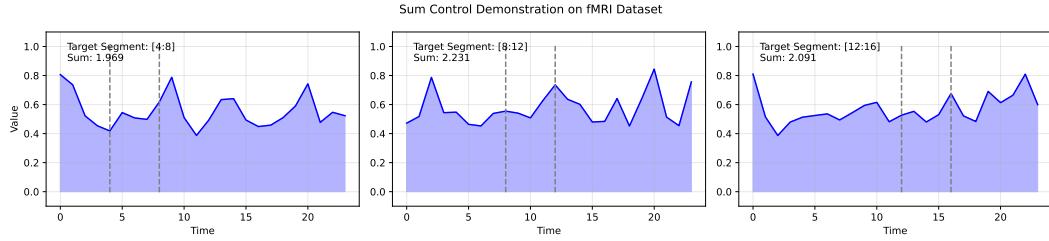


Figure 46: Demonstration of Segment-Wise Sum Control on fMRI dataset.

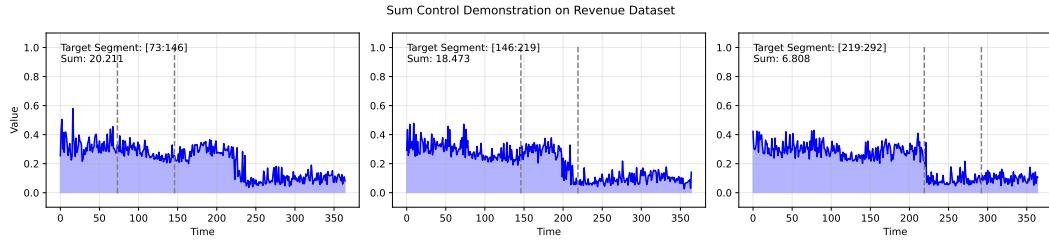


Figure 47: Demonstration of Segment-Wise Sum Control on Revenue dataset.

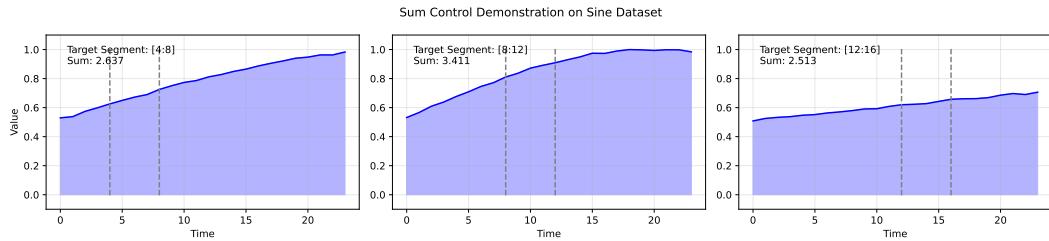


Figure 48: Demonstration of Segment-Wise Sum Control on synthetic sine wave dataset.

D.2 Kernel Density Estimate of Sum Control

Similar to the KDE analysis of point-wise control, we present KDE analysis for sum control across different datasets. The KDE peaks of controlled output (purple line) shift rightward compared to original and unconditional distributions, confirming that controlled sequences achieve higher sum values while preserving dataset-specific distributional characteristics. While this pattern does not persist consistently across different control weights and need to be further investigated.

D.2.1 KDE of Total Sum Control (Diffusion-TS)

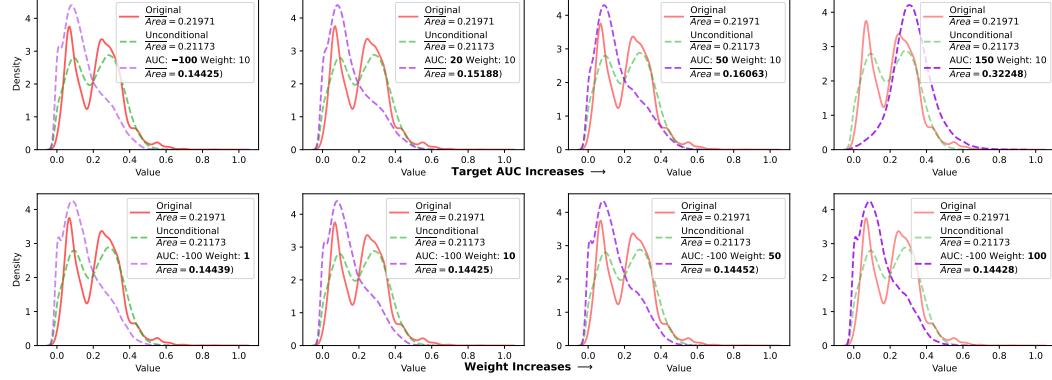


Figure 49: Kernel density estimation analysis of Revenue dataset under varying sum control targets. Top: Target analysis showing control effectiveness. Bottom: Weight analysis showing control effectiveness.

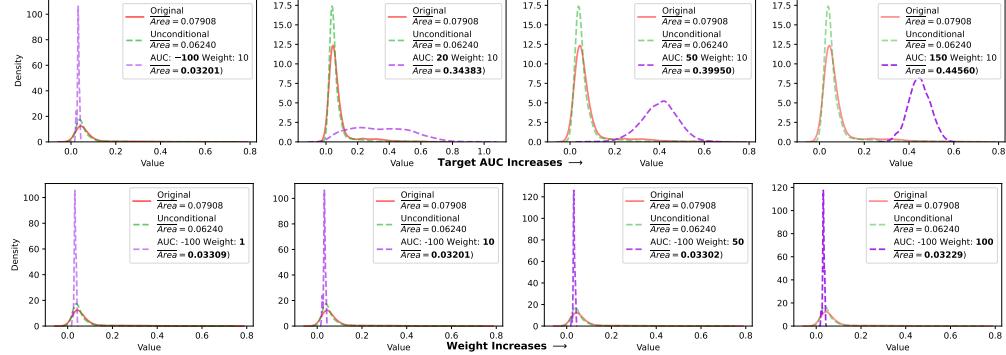


Figure 50: Kernel density estimation visualization for ETTh dataset. Top: Sum control analysis. Bottom: Weight analysis.

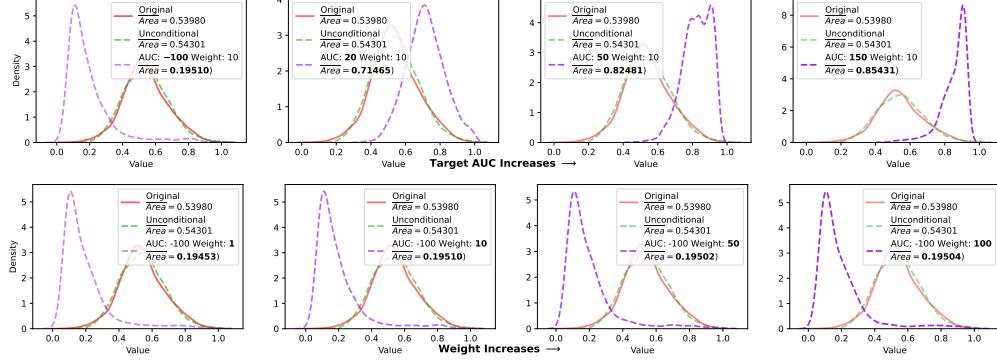


Figure 51: Kernel density estimation analysis of fMRI dataset. Top: Sum control analysis. Bottom: Weight analysis.

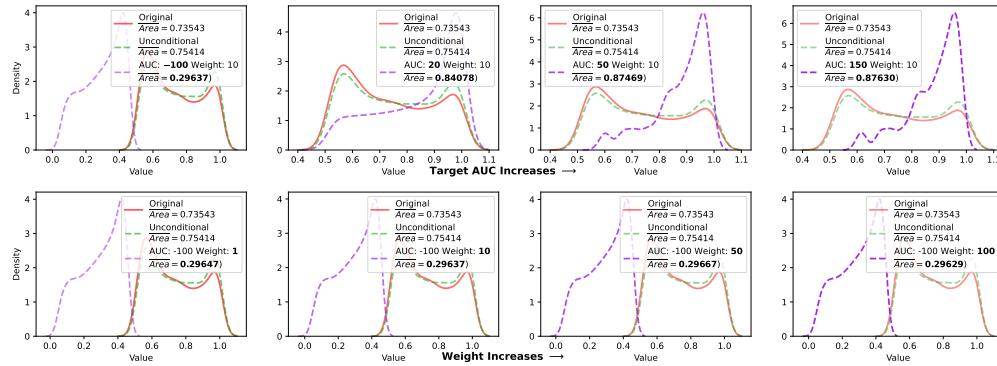


Figure 52: Kernel density estimation analysis of synthetic sine wave dataset. Top: Sum control analysis. Bottom: Weight analysis.

D.2.2 KDE of Total Sum Control (CSDI)

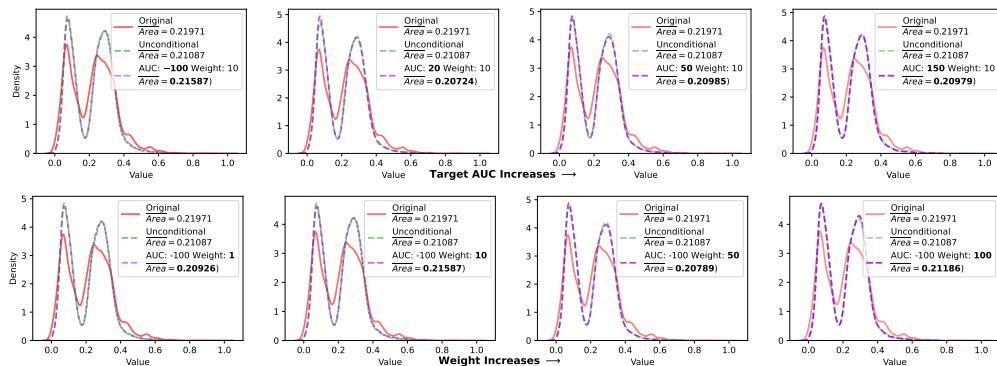


Figure 53: Kernel density estimation analysis of Revenue dataset under varying sum control targets. Top: Target analysis showing control effectiveness. Bottom: Weight analysis showing control effectiveness.

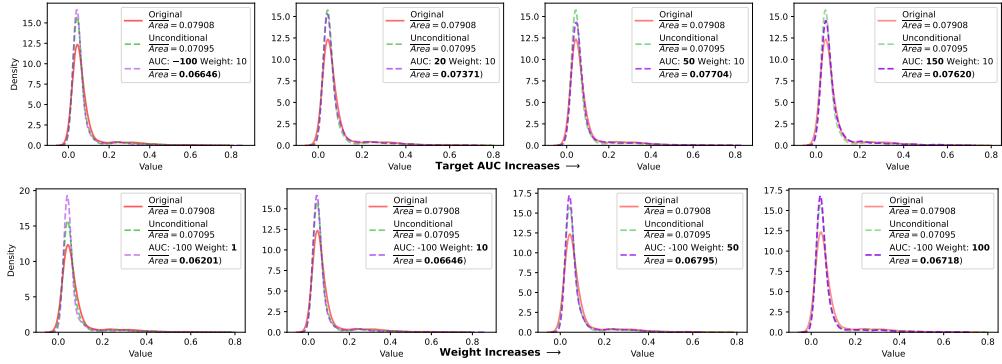


Figure 54: Kernel density estimation visualization for ETTh dataset. Top: Sum control analysis. Bottom: Weight analysis.

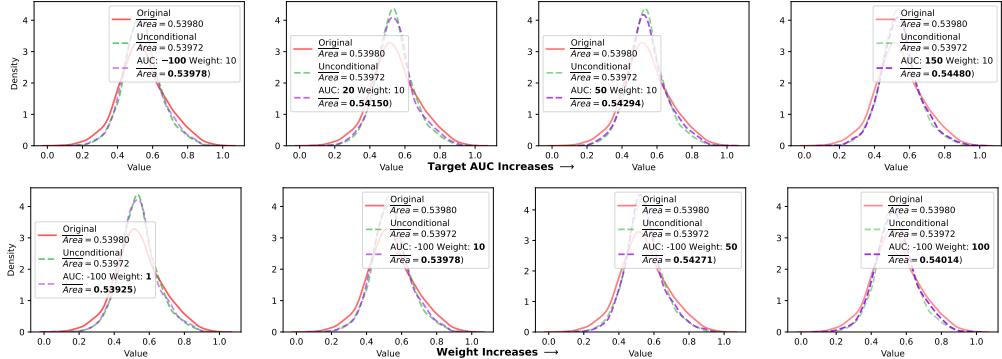


Figure 55: Kernel density estimation analysis of fMRI dataset. Top: Sum control analysis. Bottom: Weight analysis.

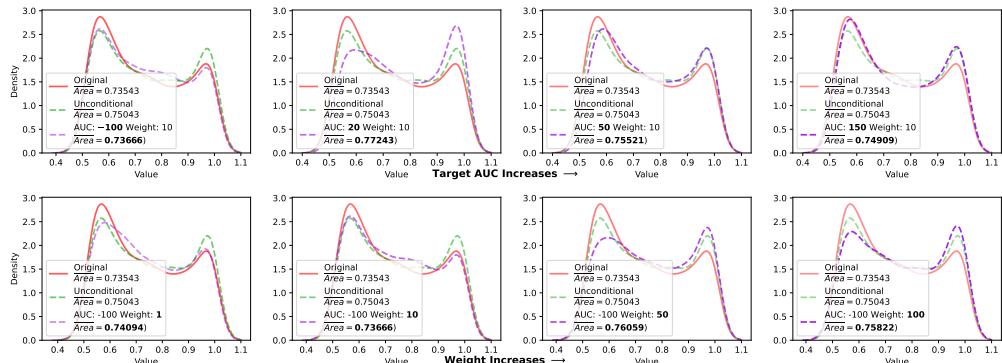


Figure 56: Kernel density estimation analysis of synthetic sine wave dataset. Top: Sum control analysis. Bottom: Weight analysis.

D.3 Averaged Sum Change Over All Segments

For the following aggregated sum change over all segments, we calculate the averaged value over segments for each dataset, with a target sum value of 150 for each segment.

D.3.1 Value Change of Segment Sum Control (Diffusion-TS)

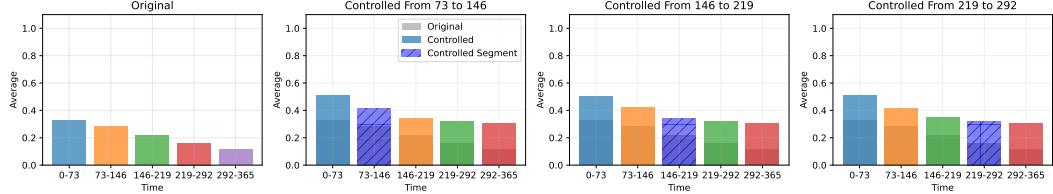


Figure 57: Segmented Summation Control on Revenue dataset.

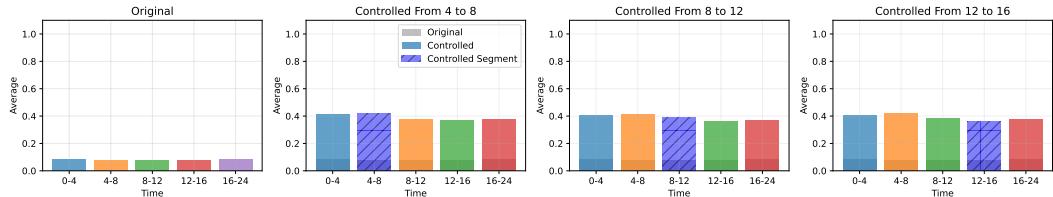


Figure 58: Segmented Summation Control on ETTh dataset.

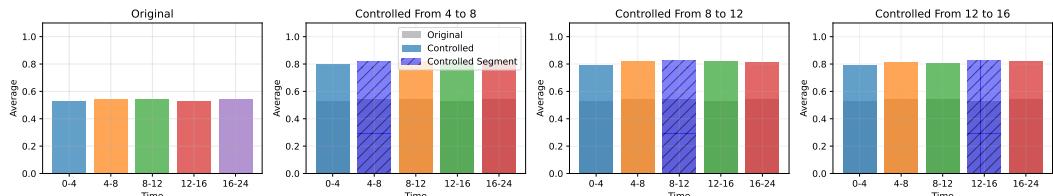


Figure 59: Segmented Summation Control on fMRI dataset.

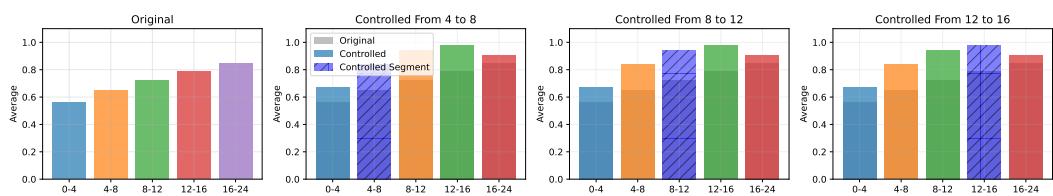


Figure 60: Segmented Summation Control on Sine dataset.

D.3.2 Value Change of Segment Sum Control (CSDI)

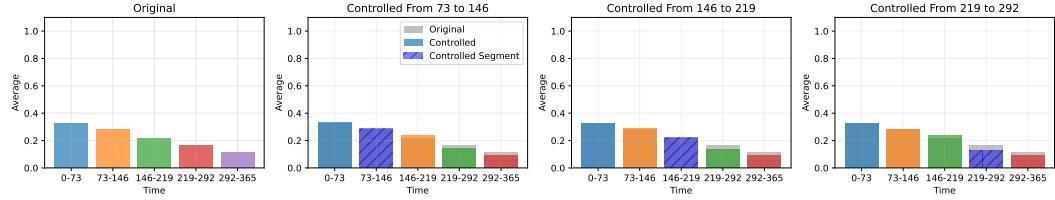


Figure 61: Segmented Summation Control on Revenue dataset.

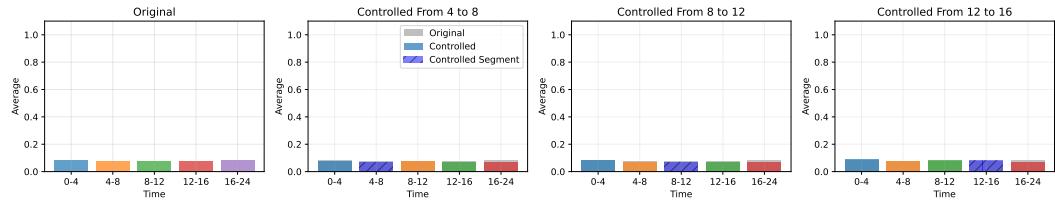


Figure 62: Segmented Summation Control on ETTh dataset.

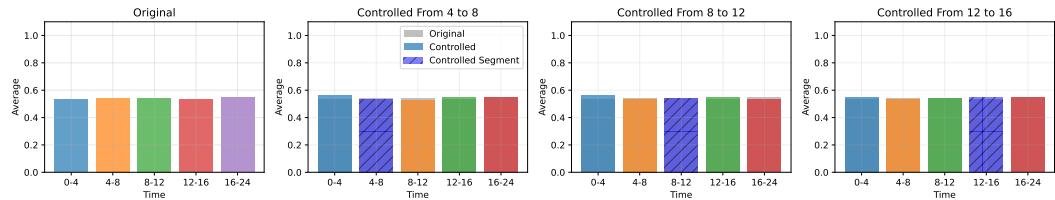


Figure 63: Segmented Summation Control on fMRI dataset.

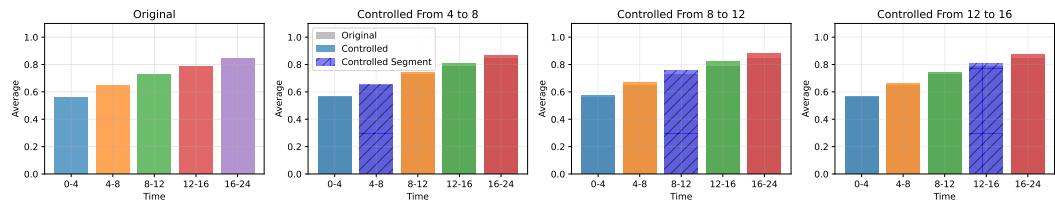


Figure 64: Segmented Summation Control on Sine dataset.

D.4 Supplement Metrics

The Discriminative, Predictive, Context-FID, and Correlational scores help quantify distribution shifts under sum control. Table D.4 demonstrates that all metrics increase significantly after applying control, indicating that the control signals effectively influence the generated time series and make them more distinguishable from the original distribution.

Table 10: Supplemental metrics for sum control performance across different datasets and target values. The results show discriminative, predictive, context-FID, and correlational scores for each dataset and control configuration. Lower scores indicate better performance.

Metrics	Control Signal	Dataset			
		ETTh	Revenue	fMRI	Sine
Discriminative Score (Lower is Better)	Unconditional	0.103±0.042	0.082±0.093	0.141±0.037	0.031±0.023
	Sum Target = 150	0.499±0.002	0.455±0.069	0.500±0.000	0.453±0.117
	Sum Target = 50	0.499±0.002	0.445±0.062	0.500±0.000	0.477±0.031
	Sum Target = 20	0.488±0.005	0.455±0.056	0.500±0.000	0.244±0.100
	Sum Target = -100	0.476±0.015	0.427±0.064	0.500±0.000	0.500±0.000
Predictive Score (Lower is Better)	Unconditional	0.256±0.002	0.065±0.026	0.103±0.002	0.094±0.000
	Sum Target = 150	0.465±0.008	0.146±0.066	0.113±0.001	0.099±0.009
	Sum Target = 50	0.418±0.007	0.107±0.006	0.108±0.001	0.108±0.040
	Sum Target = 20	0.286±0.004	0.104±0.007	0.102±0.001	0.094±0.000
	Sum Target = -100	0.289±0.005	0.106±0.009	0.116±0.003	0.111±0.008
Context-FID Score (Lower is Better)	Unconditional	0.108±0.007	1.230±0.284	0.260±0.024	0.034±0.005
	Sum Target = 150	10.608±1.651	2.648±0.927	2.740±0.480	4.346±0.818
	Sum Target = 50	8.943±1.277	2.408±0.245	2.187±0.115	2.827±0.389
	Sum Target = 20	3.129±0.473	3.533±0.849	0.614±0.111	0.884±0.281
	Sum Target = -100	5.049±0.537	5.056±0.459	3.122±0.389	19.805±1.737
Correlational Score (Lower is Better)	Unconditional	2.313±0.743	0.038±0.013	2.672±0.091	0.066±0.009
	Sum Target = 150	15.328±0.511	0.098±0.010	9.387±0.149	0.297±0.012
	Sum Target = 50	10.872±0.812	0.079±0.015	7.809±0.094	0.210±0.029
	Sum Target = 20	5.219±0.507	0.082±0.011	4.247±0.136	0.111±0.016
	Sum Target = -100	8.345±0.324	0.074±0.002	9.508±0.169	0.837±0.008

D.5 Additional Sum Control Analysis

The following figures provide comprehensive analysis of sum control performance across datasets (Unnormalized). The plots demonstrate achieved sum values compared to target values, with Original and Unconditional (Uncon) baselines as references. Analysis of control weight's impact shows minimal influence on achieved sum values across different datasets and target configurations.

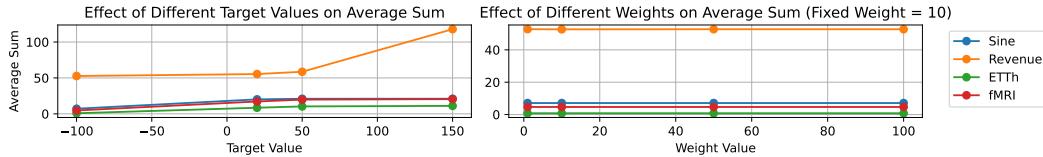


Figure 65: Comparison of achieved sum values versus target sum values across different datasets. The plots demonstrate the effectiveness of sum control guidance in reaching desired targets.

E Distribution Analysis

The FID scores reveal a fundamental trade-off between controllability and distribution preservation. While Diffusion-TS demonstrates stronger modification capabilities (FID increasing from 0.416 to 16.812 with Anchor control on fMRI), it comes at the cost of significant distribution shifts. In contrast, CSDI shows more resistance to modification but better preserves the original distribution (FID changes from 1.188 to only 1.016 under similar conditions).

This raises an important open question: How can we achieve precise temporal control while maintaining distribution fidelity? Future research should investigate mechanisms to balance these competing objectives, potentially through adaptive control strength or hybrid architectures that combine the stability of CSDI with the flexibility of Diffusion-TS.

Table 11: The complete distribution of discriminative, predictive, correlational, and FID scores for our method across different datasets and control configurations. For all metrics, lower scores indicate better performance.

Model	Metric	Control	ETTh	Revenue	fMRI	Sines
Our - CSDI	Discriminative Score	Unconditional	0.361±0.007	0.245±0.164	0.306±0.021	0.017±0.007
		Point-Wise Control	0.470±0.003	0.313±0.046	0.482±0.004	0.430±0.038
		Statistics Control	0.373±0.007	0.272±0.055	0.377±0.019	0.034±0.007
	Predictive Score	Unconditional	0.261±0.003	0.054±0.012	0.106±0.000	0.090±0.000
		Point-Wise Control	0.263±0.001	0.070±0.003	0.106±0.000	0.091±0.000
		Statistics Control	0.261±0.001	0.060±0.005	0.106±0.001	0.091±0.000
	Correlational Score	Unconditional	8.428±0.000	0.034±0.000	2.212±0.000	0.062±0.000
		Point-Wise Control	8.641±0.000	0.024±0.000	2.619±0.000	0.149±0.000
		Statistics Control	8.531±0.000	0.034±0.000	3.944±0.000	0.065±0.000
	FID Score	Unconditional	1.643±0.171	1.129±0.122	1.188±0.054	0.034±0.006
		Point-Wise Control	2.720±0.133	2.233±0.088	1.016±0.019	3.170±0.319
		Statistics Control	1.564±0.050	1.097±0.040	1.175±0.018	0.043±0.002
Our - Diffusion-TS	Discriminative Score	Unconditional	0.034±0.026	0.209±0.185	0.089±0.033	0.019±0.008
		Point-Wise Control	0.437±0.004	0.393±0.030	0.495±0.001	0.460±0.011
		Statistics Control	0.477±0.003	0.426±0.032	0.498±0.001	0.451±0.029
	Predictive Score	Unconditional	0.260±0.002	0.070±0.015	0.110±0.001	0.090±0.000
		Point-Wise Control	0.314±0.003	0.128±0.011	0.136±0.002	0.153±0.006
		Statistics Control	0.310±0.004	0.114±0.005	0.117±0.001	0.110±0.003
	Correlational Score	Unconditional	1.728±0.000	0.033±0.000	1.673±0.000	0.037±0.000
		Point-Wise Control	5.647±0.000	0.107±0.000	16.791±0.000	0.405±0.000
		Statistics Control	9.190±0.000	0.083±0.000	8.361±0.000	0.606±0.000
	FID Score	Unconditional	0.177±0.015	1.221±0.040	0.416±0.011	0.021±0.003
		Point-Wise Control	5.819±0.303	3.479±0.157	16.812±0.485	4.523±0.456
		Statistics Control	5.335±0.221	3.822±0.210	2.652±0.092	5.954±0.438

F Combined Control on Revenue Dataset

To demonstrate our method's capability to handle multiple control signals simultaneously, we present a comprehensive example using the Revenue dataset. The model successfully generates sequences that respect both anchor points and sum constraints, highlighting the flexibility and effectiveness of our approach. These generated sequences maintain the dataset's inherent distributional characteristics while precisely adhering to multiple control signals. For additional examples of combined control mechanisms, see Figure G which showcases our Time Series Editor interface in action.

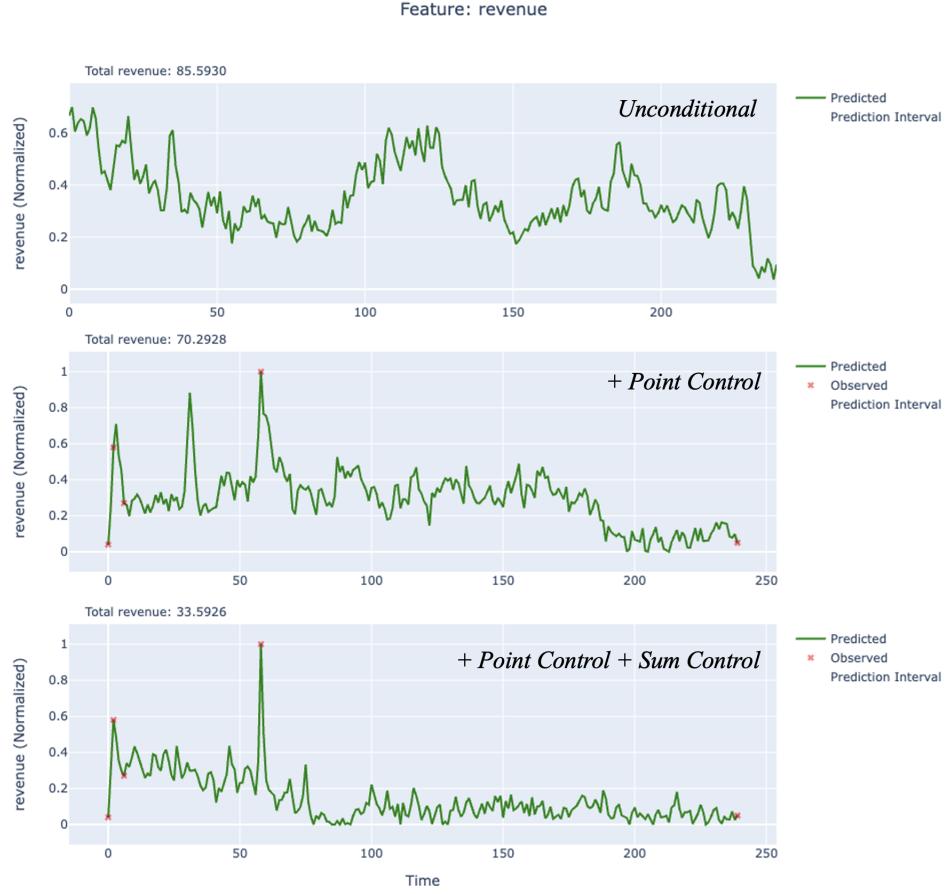


Figure 66: Demonstration of combined anchor and sum control on the Revenue dataset, showing the interaction across point-wise constraints and overall sum requirements.

G Time Series Editor

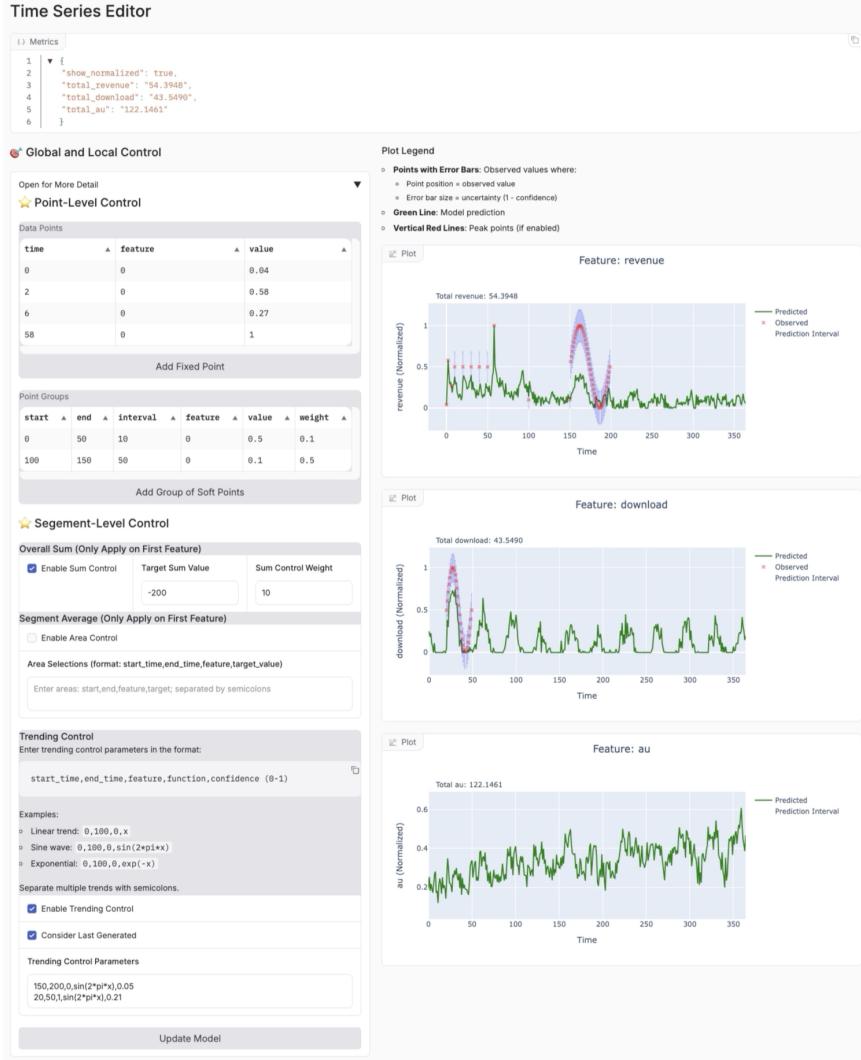


Figure 67: The Screen Shot of Time Series Editor User Interface

The Time Series Editor is designed to solve the Time Series Editing problem. It can add fixed points, soft anchors, trending control, segment-level sum, and average control. For the soft anchors, you can add anchor points based on start, end, and interval setup. The Trending Control is based on the provided function expressions with independent variable x for better precise control. We are currently developing the SketchPad mode for better user interaction, aiming to provide an All-in-One application for efficient time series editing without training.

In Figure G, the green line represents the model prediction, and the red dots represent the observed/provided anchor points. The error bar of each anchor point is the $(1 - \text{confidence})$ to demonstrate the uncertainty of the observed points. In Figure G, we use the Revenue Dataset with three features: Revenue, Download, and Daily Active Users.