ISIMUD: A UNIFIED DIFFUSION MODEL FOR TIME SERIES CAPTIONING AND GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series rarely appear in isolation in real-world, they are often accompanied by texts that provide context and expert knowledge. This naturally gives rise to two complementary cross-modality tasks: time series captioning and text-conditioned generation. Although these two tasks correspond to opposite conditional directions of the same underlying text-time series joint distribution, existing methods typically model them separately, limiting their ability to exploit shared cross-modal representations. Moreover, prior approaches are often limited to specific datasets or domains, due to the scarcity of high-quality, semantically aligned text–time series pairs. To address these challenges, we present ISIMUD, a multimodal foundation model that unifies both tasks within a shared diffusion-based architecture, enabling native modeling of both time series and text modalities. In addition, we introduce BOOK-OF-ISIMUD, a large-scale cross-domain dataset of text-time series pairs with domain-agnostic semantics, providing broad distributional coverage for training and evaluating multimodal time series models. Extensive experiments demonstrate that ISIMUD performs effectively in both in-domain and zero-shot settings, while adapting efficiently to multiple downstream tasks.

1 INTRODUCTION

Time series data play a crucial role in modern production and daily life, and are widely applied across numerous domains, including financial markets (Gao et al., 2024), healthcare (Chen et al., 2024; He et al., 2023; Jarrett et al., 2021), climate (Jing et al., 2021; 2024b), and energy systems (Gu et al., 2026). As a fundamental data modality for characterizing the dynamic evolution of systems and numerical variations over time, time series are often accompanied by textual descriptions that support human interpretation and domain-specific decision-making. For instance, physiological signals are frequently paired with diagnostic reports (Wagner et al., 2020), while meteorological measurements are often enriched with expert textual annotations (Xu et al., 2024). These accompanying texts not only enhance data representations and provide background knowledge that improves interpretability, but also help uncover the underlying mechanisms that govern the temporal dynamics.

Recent research on cross-modal interaction between time series and texts has primarily concentrated on two core tasks: time series captioning (Trabelsi et al., 2025; Xie et al., 2025) and text-conditioned generation (Ge et al., 2025; Gu et al., 2025; Li et al., 2025a). Time series captioning aims to translate complex temporal patterns into natural language that align with human cognition and domain knowledge. Conversely, text-conditioned generation seeks to synthesize time series that are semantically consistent with given textual descriptions while preserving realistic temporal structures.

Despite growing interest in these tasks, most existing studies have two major limitations: task and dataset specificity. First, although captioning and generation are dual problems that require cross-modal semantic alignment, they are usually treated as independent tasks and modeled with separate architectures. This separation prevents the model from fully exploiting the shared semantic representations across modalities (Huh et al., 2024), and increases the cost of designing, training, and deployment for task-specific systems. Second, existing methods are often tied to particular datasets, largely due to the scarcity of high-quality text–time series pairs (Li et al., 2025a; Trabelsi et al., 2025). Time series semantics vary significantly across domains; for instance, diagnostic narratives in the medical domain are not directly transferable to financial analysis. Such domain-specific semantic discrepancies make it difficult to train models that can transfer across datasets and tasks.

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.

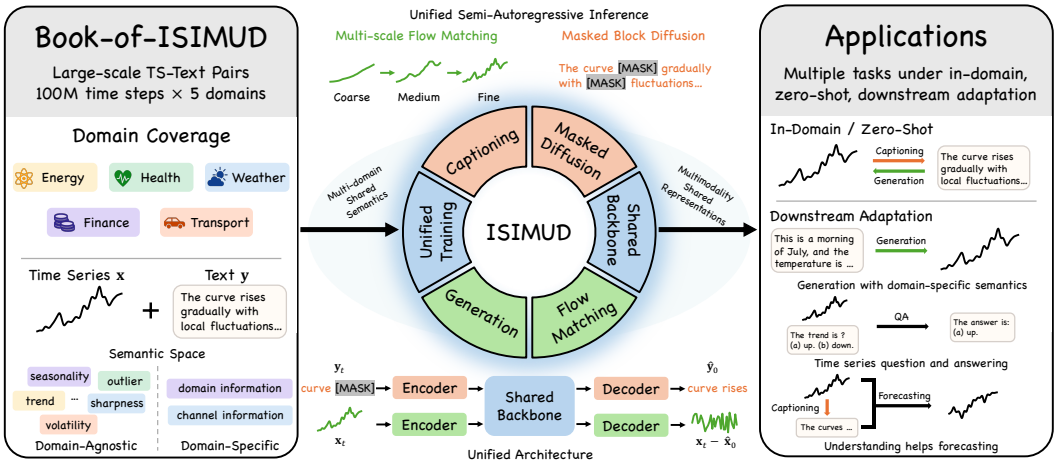

Figure 1: Overview of our contributions. **Left:** The large-scale multimodal dataset BOOK-OF-ISIMUD. **Middle:** The multimodal foundation model ISIMUD, enabling unified modeling of two tasks and modalities in a single architecture. **Right:** The applications of ISIMUD, including in-domain and zero-shot generation and captioning, as well as efficient adaptation to downstream tasks.

To overcome these limitations, we propose ISIMUD, a multimodal foundation model that unifies time series captioning and text-conditioned generation within a single generative framework. ISIMUD learns shared cross-modal representations through masked conditional generation, enabling the model to capture both directions of text-time series interaction. A key challenge lies in the modality mismatch that, text is discrete and symbolic, whereas time series are continuous and numerical. To bridge the modality gap, ISIMUD incorporates discrete (Shi et al., 2024) and continuous (Lipman et al., 2023) diffusion with shared backbone for the corresponding modalities. This design allows both modalities to be modeled natively while sharing a unified diffusion-based architecture. At inference, we further introduce a semi-autoregressive paradigm that supports efficient and high-quality captioning and generation.

Additionally, we construct BOOK-OF-ISIMUD, a large-scale, high-quality multimodal dataset for text-time series modeling. Unlike many existing datasets that are restricted to specific domains or annotation styles, BOOK-OF-ISIMUD provides text-time series pairs with domain-agnostic semantics across diverse domains, offering broad distributional coverage for training multimodal time series models. This dataset supports both large-scale pretraining, thereby facilitating the development of foundation models for time series captioning, generation, and downstream adaptation.

Our contributions are three-fold: (i) We introduce ISIMUD, a multimodal foundation model for time series captioning and generation, enabling native modeling of both modalities within a unified generative framework. (ii) We construct a large-scale multimodal dataset BOOK-OF-ISIMUD with domain-agnostic semantics, facilitating effective cross-domain learning. (iii) We establish a comprehensive benchmark for evaluating time series captioning and generation. Extensive experiments demonstrate the effectiveness of ISIMUD in in-domain and zero-shot settings, as well as its efficiency in adapting to downstream tasks, including question answering and enhancing forecasting.

2 RELATED WORK

2.1 TIME SERIES CAPTIONING

Translating complex numerical sequences into natural language is the most intuitive way for humans to understand time series. In recent years, an increasing number of works focus on time series captions. TSLM (Trabelsi et al., 2025), based on existing time series and text pairs, constructs large-scale paired datasets using in-context learning and fine-tunes a dataset-specific caption model based on a pre-trained language model. ChatTS (Xie et al., 2025) constructs synthetic datasets to fine-tune

language models, converting time series into text tokens for better understanding. TimeCAP (Lee et al., 2025) utilizes frozen LLM agents to generate text summaries to aid forecasting.

Recent studies also establish benchmarks for more rigorous evaluation of time series captioning. BEDTime (Sen et al., 2025) introduced a unified benchmark by defining three fundamental tasks: description recognition, differentiation, and open-ended generation. CaTS-Bench (Zhou et al., 2025) addressed real-world applicability by proposing context-aware captioning, which requires models to jointly reason over metadata and numerical patterns in time series. However, existing benchmarks are limited to specific datasets and fail to evaluate the generalization across different domains.

2.2 Time Series Generation

Research on time series generation has progressively evolved from unconditional to structured conditional, and more recently to unstructured conditional. Early studies primarily focused on unconditional generation, aiming to model the underlying distribution of time series without explicit control signals. Representative approaches included models based on generative adversarial networks (Li et al., 2022) and variational autoencoders (Desai et al., 2021). However, because unconditional generation offers limited controllability and struggles with precise sequence manipulation, subsequent work shifted toward controlled generation. Initial efforts in this direction introduced structured conditions, which typically take the form of discrete category labels (Lee et al., 2023) or multidimensional attributes (Jing et al., 2024a) and metadata (Narasimhan et al., 2024).

However, structured conditions are often insufficient to satisfy highly customized generation requirements. Recently, unstructured conditions have emerged as a new research focus. For instance, VerbalTS (Gu et al., 2025) was the first to systematically formalize the distinction between structured and unstructured conditions, achieving fine-grained conditional control through a multi-view noise estimator and a multi-focal text encoder. BRIDGE (Li et al., 2025a) adopts a multi-agent framework to synthesize training data and introduces a representation strategy that integrates semantic prototypes with textual embeddings. T2S (Ge et al., 2025) maps time series into a latent space via a length-adaptive variational encoder, enabling efficient generation in the latent domain. ConTSG-Bench Lan et al. (2026) is the first benchmark to comprehensively evaluate the existing time series generation methods with different type of conditions.

2.3 Multi-Task Unification in Time Series

To achieve unified and efficient modeling, many studies have explored using a single model to address multiple tasks. Early efforts primarily focused on unimodal time series tasks within a unified framework. For example, TSDiff (Kollovieh et al., 2023) employs a diffusion model with self-guidance to jointly handle unconditional generation, refinement, and forecasting of time series.

More recent work has extended this idea to multimodal settings, focusing on tasks that involve both time series and textual modalities. TimeOmni series unify question answering and forecasting within a reasoning paradigm by fine-tuning a large language model (Guan et al., 2026a) or vision language model (Guan et al., 2026b). ChatTime (Wang et al., 2025) treats time series as a foreign language for LLMs and formulates both question answering and forecasting under a next-token prediction framework. Time-MQA (Kong et al., 2025) integrates traditional numerical analysis tasks with question answering through language models. Despite these advances, existing approaches typically rely on converting time series into discrete tokens, rather than modeling in native continuous form. Furthermore, their textual outputs are often constrained to multiple-choice formats, which restricts their ability to generate rich expressions. In this work, we instead focus on two fundamental and dual tasks, generation and captioning, which remain insufficiently addressed by prior studies.

3 Method

3.1 Task Formulation

Captioning and generation are two fundamental tasks in time series analysis. They are dual to each other, representing symmetric modality transitions between time series and language. Given a paired sample consisting of a time series $\mathbf{x}$ and text $\mathbf{y}$, we formally define these two tasks as follows.

Time Series Captioning. Time series captioning aims to translate high-dimensional numerical sequences into human-readable language. Given a time series $\mathbf{x} \in \mathbb{R}^L$ with L time steps, the objective is to generate a corresponding textual description $\hat{\mathbf{y}} \in \mathbb{N}^W$ with W tokens. The generated description $\hat{\mathbf{y}}$ should accurately reflect the underlying characteristics of the time series $\mathbf{x}$, providing interpretability and helping uncover the underlying mechanisms that govern the temporal dynamics.

Text-Conditioned Time Series Generation. Time series generation focuses on synthesizing time series conditioned on natural language descriptions. Given a text $\mathbf{y} \in \mathbb{N}^W$ with W tokens, the goal is to produce a corresponding time series $\hat{\mathbf{x}} \in \mathbb{R}^L$ with L time steps. The generated sequence $\hat{\mathbf{x}}$ should exhibit high fidelity and be semantically consistent with the information conveyed in the textual description $\mathbf{y}$.

3.2 ISIMUD: UNIFIED MULTIMODAL MODEL FOR TIME SERIES CAPTIONING AND GENERATION

Unified modeling across tasks has recently gained significant attention as an efficient and mutually beneficial paradigm (Zhao et al., 2026). However, applying this paradigm to time series presents two key challenges: the modality gap and the task gap. Specifically, it is non-trivial to design a unified representation within a single architecture that can simultaneously accommodate discrete textual tokens and continuous time series values. Moreover, generation and captioning involve inherently different objectives: the former prioritizes accurate reconstruction of time series, while the latter emphasizes semantic understanding and linguistic coherence. This discrepancy makes it difficult to balance performance across tasks within a shared training framework. In this work, we formulate time series captioning and generation as a unified masked conditional generation problem, where each task corresponds to generating a target modality from corrupted inputs conditioned on the other modality. We propose a unified model and training pipeline that supports both tasks while preserving the native modeling of each modality, avoiding the mismatch between the discrete and continuous representations, leading to improved performance in both tasks.

3.2.1 UNIFIED ARCHITECTURE WITH SEMI-AUTOREGRESSIVE INFERENCE

We construct a unified model f_θ with a shared diffusion language model (Cheng et al., 2025) backbone, complemented by modality-specific encoders and decoders. To support native modeling of both modalities, we adopt continuous diffusion for time series and discrete diffusion for texts, with details explained in Appendix B. We adopt a unified *semi-autoregressive inference* for both captioning and generation, as illustrated in Fig. 2. Specifically, we partition both text and time series into non-overlapping blocks (or scales). Parallel diffusion is performed within each block, while autoregressive generation operates across blocks. For captioning, we adopt a next-block prediction strategy, which has been shown to be an efficient paradigm for text generation in diffusion language models (Arriola et al., 2025). For generation, we propose a novel next-scale prediction strategy that enables progressive refinement from coarse to fine-grained, resulting in higher-quality generation.

Next-Block Prediction. For the captioning task, we partition the text into multiple blocks based on temporal dependencies. Specifically, the textual description $\mathbf{y} \in \mathbb{N}^W$ is divided into L_B^{text} blocks $\{\mathbf{y}_i\}_{i=1}^{L_B^{\text{text}}}$, where $\mathbf{y}_i \in \mathbb{N}^{B^{\text{text}}}$ and B^{text} is the block size. End tokens are appended to $\mathbf{y}$ to ensure that $W = L_B^{\text{text}} \times B^{\text{text}}$. Each textual block is independently perturbed via random masking, resulting in noisy blocks $\{\mathbf{y}_{i,t}\}_{i=1}^{L_B^{\text{text}}}$, where t denotes the diffusion step. We adopt the original tokenizer and word embedding layers from the diffusion language model (Cheng et al., 2025) to encode the textual sequence $\mathbf{y}_t \in \mathbb{N}^{L_B^{\text{text}} \times B^{\text{text}}}$ into hidden representations $\mathbf{b}_t^{\text{text}} \in \mathbb{R}^{L_B^{\text{text}} \times B^{\text{text}} \times D}$, where D is the hidden dimension. As shown in the right of Fig. 2, each text block $\mathbf{y}_{i,t} \in \mathbb{R}^{B^{\text{text}} \times D}$, diffusion is performed in parallel, while autoregressive generation is applied across the sequence of blocks $\{\mathbf{y}_{i,t}\}_{i=1}^{L_B^{\text{text}}}$.

Next-Scale Prediction. For the generation task, we heuristically construct time series $\mathbf{x} \in \mathbb{R}^L$ at multiple scales using moving average. Specifically, the time series $\mathbf{x}$ is independently transformed into L_B^{ts} sequences at different scales $\{\mathbf{x}_i\}_{i=1}^{L_B^{\text{ts}}}$, where a larger scale represents more details in time series. $\mathbf{x}_i = \text{MA}(\mathbf{x}, k^{L_B^{\text{ts}}-i})$, $\text{MA}(\cdot,\cdot)$ denotes the moving average function with details in Appendix F.1, k is the base kernel size, and $k^{L_B^{\text{ts}}-i}$ is the corresponding window size for each scale. The time series at each scale is independently perturbed through the forward process of flow match-

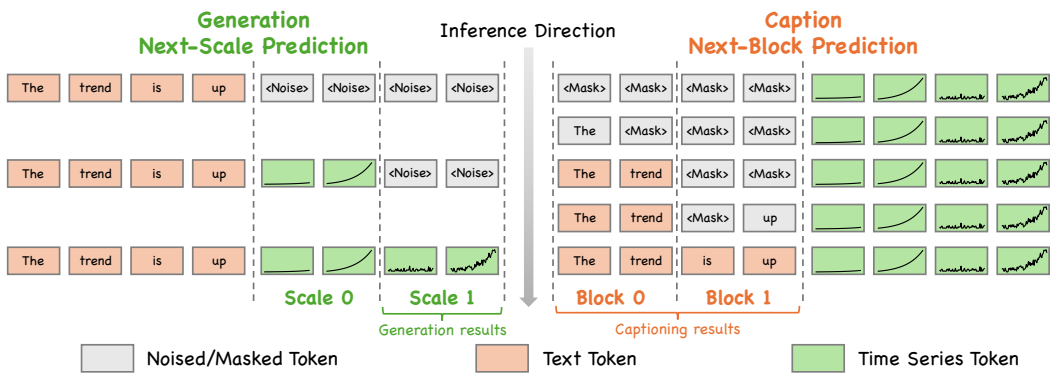

Figure 2: The unified semi-autoregressive inference paradigm for captioning and generation.

ing (Lipman et al., 2023) to obtain the noisy version $\{\mathbf{x}_{i,t}\}_{i=1}^{L_B^{\text{ts}}}$, where t denotes the diffusion step. We employ a patch encoder to map the time series at different scales $\mathbf{x}_{i,t} \in \mathbb{R}^L$ into hidden embeddings $\mathbf{b}_{i,t}^{\text{ts}} \in \mathbb{R}^{B^{\text{ts}} \times D}$, where $B^{\text{ts}} = \lfloor \frac{L}{P} \rfloor$ is the number of patches, P is the patch size, and D is the hidden dimension. Parallel diffusion is applied within each time series scale $\mathbf{x}_{i,t} \in \mathbb{R}^{B^{\text{ts}} \times D}$, while autoregressive generation is performed across the scales $\{\mathbf{x}_{i,t}\}_{i=1}^{L_B^{\text{ts}}}$.

The inference process is formulated as a unified semi-autoregressive paradigm with task-specific prompts p^{gen} for generation and p^{cap} for captioning:

$$P(\hat{\mathbf{x}}_{i,t}) \sim f_\theta(p^{\text{gen}}, \mathbf{x}_{i,t+1}, \mathbf{x}_{<i,0}, \mathbf{y}_{\le L_B^{\text{text}},0}), P(\hat{\mathbf{y}}_{i,t}) \sim f_\theta(p^{\text{cap}}, \mathbf{y}_{i,t+1}, \mathbf{y}_{<i,0}, \mathbf{x}_{L_B^{\text{ts}},0}). \tag{1}$$

Discussion. Compared with existing unified models for multimodal time series (Guan et al., 2026a; Wang et al., 2025), ISIMUD is the first to explore using the diffusion model as backbone and support the native modeling of both time series and text modalities. We introduce a novel next-scale prediction mechanism for time series generation within diffusion language models, facilitating fine-grained control. With the next-block prediction mechanism, we establish a unified semi-autoregressive framework that enhances the learning of cohesive multimodal representations. Experiments in Sec. 4.4 demonstrate that this design yields superior generation and captioning performance.

3.2.2 Unified Training With Masked-Conditional Generation Paradigm

As the two foundational tasks in time series analysis, captioning and generation can be viewed as dual tasks for each other, sharing the target of learning the bidirectional transformation. Works (Zhao et al., 2026; Li et al., 2025b) have proven that training the unified model with similar tasks will bring mutual benefits. To achieve efficient unified training across the tasks, we model the training of the time series captioning and generation into a unified *masked-conditional generation* paradigm, which can be viewed as maximizing the likelihood of generating a target modality from masked or corrupted states conditioned on the other clean modality.

We first perturb the multi-scale time series $\{\mathbf{x}_i\}$ through adding Gaussian noise and texts $\{\mathbf{y}_i\}$ through random masking, following the traditional strategy of continuous and discrete diffusion, as discussed in Appendix B. We concatenate both the original and perturbed sequences of two modalities $\{\mathbf{x}_{i,0}, \mathbf{x}_{i,t}\}_{i=1}^{L_B^{\text{ts}}}, \{\mathbf{y}_{i,0}, \mathbf{y}_{i,t}\}_{i=1}^{L_B^{\text{text}}}$ and feed them into the model f_θ to calculate the loss:

$$\begin{aligned}
\mathcal{L}_{\text{gen}}(\theta) &= \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[-\sum_{i=1}^{L_B^{\text{ts}}} \log f_\theta(\mathbf{x}_{i,0}|\mathbf{x}_{i,t}, \mathbf{x}_{<i,0}, \mathbf{y}_{\le L_B^{\text{text}},0})\right], \\
\mathcal{L}_{\text{cap}}(\theta) &= \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[-\sum_{i=1}^{L_B^{\text{text}}} \log f_\theta(\mathbf{y}_{i,0}|\mathbf{y}_{i,t}, \mathbf{y}_{<i,0}, \mathbf{x}_{L_B^{\text{ts}},0})\right], \\
\mathcal{L}(\theta) &= \lambda_{\text{gen}} \times \mathcal{L}_{\text{gen}}(\theta) + \lambda_{\text{cap}} \times \mathcal{L}_{\text{cap}}(\theta),
\end{aligned} \tag{2}$$

where $\lambda_{\text{gen}}, \lambda_{\text{cap}}$ are the loss weight to balance the performance across two tasks.

Discussion. To the best of our knowledge, ISIMUD is the first work that unifies time series captioning and generation within a single framework. In contrast to existing works (Guan et al., 2026a), which unify time series question answering (QA) and forecasting, the tasks we consider are intuitively dual and mutually benefit each other. We further demonstrate the mutual benefits of unifying these two tasks in Sec. 4.4, whereas related work only prove that time series QA can improve forecasting.

3.3 BOOK-OF-ISIMUD: MULTIMODAL DATASET WITH DOMAIN-AGNOSTIC SEMANTICS

To construct a foundation model for time series captioning and generation, large-scale and high-quality paired data consisting of text and time series is essential for training. However, such paired data are scarce in real-world scenarios, and the semantics of text can vary significantly across different domains. To address these challenges, we propose BOOK-OF-ISIMUD, a multimodal dataset that provides domain-agnostic semantics for time series across multiple domains.

Specifically, we define domain-agnostic semantics by how humans interpret unfamiliar time series, typically focusing on the statistical and morphological attributes. We collect a large-scale time series samples totaling 100 million time steps across five domains: energy, health, weather, finance, and transport. We further invite the human experts to identify significant and common attributes. Detailed descriptions of the datasets are provided in Appendix C.1. To ensure that the textual descriptions comprehensively and accurately capture the features of time series while maintaining diversity and remaining aligned with natural language, we propose an agent-collaborative textual annotation pipeline including: a describer D and a refiner R, both built upon LLM Qwen3 (Yang et al., 2025), as illustrated in Fig. 3.

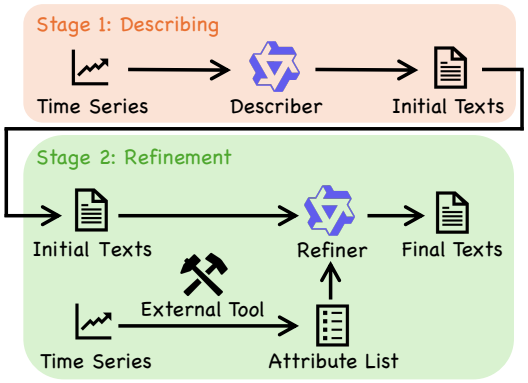

Figure 3: The textual annotation pipeline consist of two stages: describing and refinement. The describer is responsible to generate initial descriptions given the time series, while the refiner refine the texts through the extracted attributes.

Stage 1. Describing. The goal of the describer is to generate initial textual descriptions for numerical sequences. Given a univariate time series $\mathbf{x} \in \mathbb{R}^L$ of L time steps and an instruction prompt p_1 specifying the attributes to be considered, the describer D produces the corresponding initial description $\tilde{\mathbf{y}} = D(\mathbf{x}, p_1)$. The describer is instructed to generate diverse descriptions that are as close to natural language as possible. However, large language models often struggle to interpret numerical sequences, causing the initial descriptions to inaccurately capture the complex attributes of the time series. Consequently, the initial descriptions $\tilde{\mathbf{y}}$ require further refinement.

Stage 2. Refinement. The refiner's goal is to revise the initial textual descriptions $\tilde{\mathbf{y}}$. Since large language models often fail to accurately capture the complex attributes of time series, we utilize the external tool to extract these attributes. Specifically, we use *tsfresh* (Christ et al., 2018) to extract the attributes in numerical form and transfer them to textual prompts p_2, which include precise descriptions of the attributes along with instructions for refinement. Finally, we prompt the refiner to generate the final textual description $\mathbf{y} = R(\tilde{\mathbf{y}}, p_2)$. Given the initial description and the refinement instructions, the refiner revises the descriptions of specific attributes to enhance their accuracy.

In summary, the describer generates diverse descriptions of time series that closely resemble human language, while the refiner ensures the descriptions are accurate. The implement details of the annotation pipelines are provided in Appendix D.1. We further evaluate the quality of the annotated descriptions by human and show the necessity of the agent-collaborative design in Appendix D.2.

In comparison to existing multimodal time series datasets (Guan et al., 2026a), BOOK-OF-ISIMUD is the first large-scale text–time series paired dataset designed specifically for the dual tasks of generation and captioning. It offers more comprehensive domain and sample coverage than existing generation dataset (Gu et al., 2025), thereby providing essential data for pretraining foundation models in generation and captioning. We define the domain-agnostic semantics for unification across different domains, which enables efficient data annotation without requiring expert knowledge. Furthermore, the agent-collaborative annotation pipeline ensures both diversity and accuracy in the texts.

Table 1: The performance of generation and captioning on five in-domain datasets. All methods are trained on BOOK-OF-ISIMUD and evaluated on the test split of each dataset. Arrows ↑ (↓) indicate that higher (lower) values are better. Best results are in bold and second-best results are underlined.

Dataset		era5		ercot		city_bikes		covid_deaths		favorita_transactions	
Generation											
Method	**Evaluation Type**	↓ J-FTSD	↑ CTTP	↓ J-FTSD	↑ CTTP	↓ J-FTSD	↑ CTTP	↓ J-FTSD	↑ CTTP	↓ J-FTSD	↑ CTTP
T2S (Ge et al., 2025)	In-domain	$1.685_{\pm 0.135}$	$0.113_{\pm 0.020}$	$2.115_{\pm 0.133}$	$0.089_{\pm 0.006}$	$1.731_{\pm 0.032}$	$0.182_{\pm 0.022}$	$2.190_{\pm 0.036}$	$0.194_{\pm 0.037}$	$2.183_{\pm 0.026}$	$0.453_{\pm 0.069}$
Bridge (Li et al., 2025a)	In-domain	$1.581_{\pm 0.023}$	$0.417_{\pm 0.012}$	$1.650_{\pm 0.080}$	$0.414_{\pm 0.062}$	$1.078_{\pm 0.017}$	$0.322_{\pm 0.019}$	$2.092_{\pm 0.040}$	$0.530_{\pm 0.054}$	$2.125_{\pm 0.160}$	$0.540_{\pm 0.014}$
VerbalTS (Gu et al., 2025)	In-domain	**1.169**$_{\pm 0.003}$	$\underline{0.451}_{\pm 0.034}$	**1.439**$_{\pm 0.045}$	$\underline{0.566}_{\pm 0.008}$	**0.860**$_{\pm 0.011}$	$\underline{0.442}_{\pm 0.037}$	$\underline{2.013}_{\pm 0.021}$	$\underline{0.583}_{\pm 0.003}$	$\underline{1.787}_{\pm 0.012}$	$0.582_{\pm 0.004}$
DiffuSETS (Lai et al., 2025)	In-domain	$1.921_{\pm 0.212}$	$0.092_{\pm 0.047}$	$1.937_{\pm 0.178}$	$0.383_{\pm 0.172}$	$1.123_{\pm 0.019}$	$0.213_{\pm 0.009}$	$2.059_{\pm 0.074}$	$0.570_{\pm 0.023}$	$1.906_{\pm 0.012}$	$\underline{0.592}_{\pm 0.004}$
Text2Motion (Guo et al., 2022)	In-domain	$1.789_{\pm 0.034}$	$0.116_{\pm 0.017}$	$1.763_{\pm 0.147}$	$0.343_{\pm 0.136}$	$1.192_{\pm 0.134}$	$0.261_{\pm 0.032}$	$2.227_{\pm 0.010}$	$0.427_{\pm 0.029}$	$2.094_{\pm 0.216}$	$0.477_{\pm 0.076}$
ISIMUD (ours)	In-domain	$\underline{1.450}_{\pm 0.001}$	**0.622**$_{\pm 0.001}$	$\underline{1.478}_{\pm 0.001}$	**0.598**$_{\pm 0.000}$	$\underline{0.866}_{\pm 0.001}$	**0.544**$_{\pm 0.001}$	**1.981**$_{\pm 0.002}$	**0.602**$_{\pm 0.001}$	**1.741**$_{\pm 0.013}$	**0.638**$_{\pm 0.002}$
Captioning											
Method	**Evaluation Type**	↑ ROUGE-L	↑ CTTP	↑ ROUGE-L	↑ CTTP	↑ ROUGE-L	↑ CTTP	↑ ROUGE-L	↑ CTTP	↑ ROUGE-L	↑ CTTP
TSLM (Trabelsi et al., 2025)	In-domain	$\underline{0.342}_{\pm 0.000}$	$\underline{0.405}_{\pm 0.000}$	$\underline{0.362}_{\pm 0.000}$	$\underline{0.516}_{\pm 0.000}$	$\underline{0.348}_{\pm 0.000}$	$\underline{0.487}_{\pm 0.000}$	$\underline{0.377}_{\pm 0.000}$	$\underline{0.512}_{\pm 0.000}$	$\underline{0.319}_{\pm 0.000}$	$\underline{0.547}_{\pm 0.000}$
ChatTS (Xie et al., 2025)	In-domain	$0.312_{\pm 0.101}$	$0.389_{\pm 0.023}$	$0.330_{\pm 0.079}$	$0.425_{\pm 0.007}$	$0.319_{\pm 0.065}$	$0.416_{\pm 0.011}$	$0.306_{\pm 0.034}$	$0.476_{\pm 0.045}$	$0.290_{\pm 0.009}$	$0.433_{\pm 0.060}$
ChatTime (Wang et al., 2025)	In-domain	$0.281_{\pm 0.042}$	$0.182_{\pm 0.131}$	$0.279_{\pm 0.057}$	$0.299_{\pm 0.128}$	$0.266_{\pm 0.046}$	$0.209_{\pm 0.027}$	$0.289_{\pm 0.035}$	$0.239_{\pm 0.041}$	$0.272_{\pm 0.022}$	$0.151_{\pm 0.036}$
OpenTSLM (Langer et al., 2026)	In-domain	$0.134_{\pm 0.036}$	$0.128_{\pm 0.010}$	$0.168_{\pm 0.039}$	$0.023_{\pm 0.018}$	$0.122_{\pm 0.055}$	$0.158_{\pm 0.011}$	$0.132_{\pm 0.044}$	$0.025_{\pm 0.012}$	$0.115_{\pm 0.057}$	$0.059_{\pm 0.009}$
ISIMUD (ours)	In-domain	**0.485**$_{\pm 0.001}$	**0.617**$_{\pm 0.001}$	**0.473**$_{\pm 0.000}$	**0.581**$_{\pm 0.000}$	**0.430**$_{\pm 0.001}$	**0.594**$_{\pm 0.001}$	**0.474**$_{\pm 0.000}$	**0.620**$_{\pm 0.000}$	**0.415**$_{\pm 0.001}$	**0.628**$_{\pm 0.000}$

4 EXPERIMENT

In this section, we provide the experimental setup and results, structured around the following research questions: **RQ1 (Unification):** Can ISIMUD effectively handle both generation and captioning tasks within a single model architecture? **RQ2 (Generalization):** Can the generation and captioning capabilities of ISIMUD generalize to unseen datasets? **RQ3 (Adaptation):** Can ISIMUD adapt to different downstream tasks at a low cost? The code will be available upon the acceptance.

4.1 EXPERIMENTAL SETUP

Datasets. We utilize three categories of datasets for evaluation: (i) In-domain datasets. These include **era5**, **ercot**, **city_bikes**, **covid_deaths**, and **favorita_transactions**, whose training splits are included in BOOK-OF-ISIMUD. (ii) Zero-shot datasets. These include **weatherbench** and **fox**, which are not contained in BOOK-OF-ISIMUD but share similar domain-agnostic semantics. (iii) Downstream datasets. For generation, we consider the datasets from ConTSG-Bench (Lan et al., 2026), whose semantics differ from those used in pretraining. For question answering, we include the datasets from ChatTime (Wang et al., 2025). Details of datasets are provided in Appendix C.

Evaluation Metrics. For the generation, **J-FTSD** (Narasimhan et al., 2024) is reported to quantify the discrepancy between the distribution of generated and real data. **CTTP** score (Gu et al., 2025) is incorporated to measure the semantic consistency between generated time series and texts. For the captioning, we consider **ROUGE-L** (Lin, 2004) to evaluate the textual structural coherence, and incorporate the **CTTP** score to measure the semantic consistency between the caption results and the time series. The details of the metrics are provided in Appendix E.

Baselines. For generation, we consider dataset-specific methods for text-conditioned time series generation: **VerbalTS** (Gu et al., 2025), **BRIDGE** (Li et al., 2025a), **T2S** (Ge et al., 2025), **DiffuSETS** (Lai et al., 2025), and **Text2Motion** (Guo et al., 2022). For captioning, we consider the foundation model **ChatTS** (Xie et al., 2025), **ChatTime** (Wang et al., 2025) and the dataset-specific method **TSLM** (Trabelsi et al., 2025), **OpenTSLM** (Langer et al., 2026). We report the full-shot performance of dataset-specific methods and evaluate the in-domain and zero-shot capabilities of foundation models. For downstream tasks, we follow the baselines reported in the corresponding benchmark with details of evaluation protocols in Appendix F.3

Setting. ISIMUD is pretrained on the BOOK-OF-ISIMUD with the optimization target of both generation and captioning. For downstream evaluation, we leverage the training data of the original dataset to finetune ISIMUD. Details of implementation and training can be found in Appendix F.2.

4.2 IN-DOMAIN EVALUATION

In this section, we quantitatively evaluate both the generation and captioning performance of ISIMUD on in-domain datasets, where the training splits of each dataset are included in the pretraining cor-

pus BOOK-OF-ISIMUD. The results are summarized in Tab. 1. For the generation task, ISIMUD achieves strong performance in terms of both fidelity and semantic consistency, consistently ranking first or second across different datasets. For the captioning task, ISIMUD demonstrates superior performance in both structural coherence and semantic consistency of the generated text, exhibiting clear advantages over baseline methods. Overall, these results address **RQ1**, showing that ISIMUD *can effectively handle both tasks within a unified architecture.*

4.3 ZERO-SHOT EVALUATION

In this section, we quantitatively evaluate the generation and captioning performance of ISIMUD on zero-shot datasets. These datasets are not included in the pretraining corpus of BOOK-OF-ISIMUD but share domain-agnostic semantics in the textual descriptions. The results are presented in Tab. 2. Compared with full-shot baselines, our method achieves competitive performance on these unseen datasets. This demonstrates that the domain-agnostic semantics we define generalize effectively across different datasets, and our method is capable of capturing these features. Furthermore, under the same zero-shot setting, our method exhibits stronger generalization to unseen data compared with other baselines. This addresses **RQ2**, showing that ISIMUD *generalizes to unseen datasets.*

Table 2: The performance of generation and caption on zero-shot datasets. "Full-shot" represents the methods that are trained on the train split of the datasets and evaluated on the test split, "Zero-shot" represents the methods that are never trained on the corresponding datasets.

Dataset		weatherbench		fox	
Generation					
Method	**Evaluation Type**	↓ J-FTSD	↑ CTTP	↓ J-FTSD	↑ CTTP
T2S (Ge et al., 2025)	Full-shot	$1.825_{\pm 0.039}$	$0.380_{\pm 0.036}$	$2.104_{\pm 0.015}$	$0.265_{\pm 0.036}$
Bridge (Li et al., 2025a)	Full-shot	$1.325_{\pm 0.014}$	$\underline{0.474}_{\pm 0.003}$	$1.314_{\pm 0.011}$	$0.383_{\pm 0.057}$
VerbalTS (Gu et al., 2025)	Full-shot	$\underline{1.176}_{\pm 0.004}$	$0.472_{\pm 0.003}$	$\underline{1.134}_{\pm 0.005}$	$0.472_{\pm 0.001}$
DiffuSETS (Lai et al., 2025)	Full-shot	$1.319_{\pm 0.057}$	$0.456_{\pm 0.068}$	$1.364_{\pm 0.024}$	$\underline{0.485}_{\pm 0.008}$
Text2Motion (Guo et al., 2022)	Full-shot	$1.378_{\pm 0.007}$	$0.435_{\pm 0.009}$	$1.451_{\pm 0.016}$	$0.310_{\pm 0.009}$
ISIMUD (ours)	Zero-shot	$\mathbf{0.972}_{\pm 0.008}$	$\mathbf{0.490}_{\pm 0.002}$	$\mathbf{1.096}_{\pm 0.002}$	$\mathbf{0.501}_{\pm 0.001}$
Captioning					
Method	**Evaluation Type**	↑ ROUGE-L	↑ CTTP	↑ ROUGE-L	↑ CTTP
TSLM (Trabelsi et al., 2025)	Full-shot	$\underline{0.299}_{\pm 0.000}$	$\underline{0.396}_{\pm 0.000}$	$\underline{0.342}_{\pm 0.000}$	$\underline{0.433}_{\pm 0.000}$
ChatTS (Xie et al., 2025)	Zero-shot	$0.280_{\pm 0.036}$	$0.307_{\pm 0.013}$	$0.290_{\pm 0.012}$	$0.140_{\pm 0.055}$
ChatTime (Wang et al., 2025)	Zero-shot	$0.284_{\pm 0.058}$	$0.211_{\pm 0.033}$	$0.275_{\pm 0.029}$	$0.202_{\pm 0.078}$
OpenTSLM (Langer et al., 2026)	Zero-shot	$0.131_{\pm 0.066}$	$0.147_{\pm 0.016}$	$0.142_{\pm 0.070}$	$0.087_{\pm 0.010}$
ISIMUD (ours)	Zero-shot	$\mathbf{0.386}_{\pm 0.000}$	$\mathbf{0.438}_{\pm 0.000}$	$\mathbf{0.422}_{\pm 0.000}$	$\mathbf{0.518}_{\pm 0.000}$

4.4 ABLATION STUDY

In this section, we analyze the advantages of unified training and the improvements brought by the semi-autoregressive inference paradigm. Specifically, we first ablate the unified training strategy by training two separate and independent models. As shown in Tab. 3, both generation and captioning performance degrade without unified training, suggesting that *jointly modeling the dual tasks leads to mutual benefits and improved overall performance.* We further ablate the design of the semi-autoregressive inference, including next-block prediction (NBP) for captioning and next-scale prediction (NSP) for generation. The results demonstrate that the semi-autoregressive paradigm is effective and efficient: *NBP improves the logical coherence of captioning, while NSP enables more fine-grained and high-quality generation.* We further conduct significance tests in Appendix H.2.

Table 3: The ablation study of ISIMUD, where "NBP" represents the next-block prediction paradigm of the caption task, "Caption" represents the caption training, "NSP" represents the next-scale prediction paradigm of the generation task. "Generation" represents the generation training.

Dataset	**ercot** (in-domain)				**fox** (zero-shot)			
	↓ J-FTSD	↑ CTTP	↑ ROUGE-L	↑ CTTP	↓ J-FTSD	↑ CTTP	↑ ROUGE-L	↑ CTTP
ISIMUD (ours)	$\mathbf{1.478}_{\pm 0.001}$	$\mathbf{0.598}_{\pm 0.000}$	$\mathbf{0.473}_{\pm 0.000}$	$\mathbf{0.581}_{\pm 0.000}$	$\mathbf{1.096}_{\pm 0.002}$	$\mathbf{0.501}_{\pm 0.001}$	$\mathbf{0.422}_{\pm 0.000}$	$\mathbf{0.518}_{\pm 0.000}$
w/o NBP	$1.492_{\pm 0.001}$	$0.563_{\pm 0.001}$	$0.266_{\pm 0.000}$	$0.144_{\pm 0.000}$	$1.106_{\pm 0.001}$	$0.461_{\pm 0.001}$	$0.255_{\pm 0.000}$	$0.068_{\pm 0.000}$
w/o NSP	$1.489_{\pm 0.001}$	$0.597_{\pm 0.001}$	$0.467_{\pm 0.000}$	$0.571_{\pm 0.000}$	$1.124_{\pm 0.002}$	$0.489_{\pm 0.003}$	$0.420_{\pm 0.000}$	$0.511_{\pm 0.000}$
w/o Caption	$1.494_{\pm 0.001}$	$0.593_{\pm 0.000}$	-	-	$1.102_{\pm 0.001}$	$0.495_{\pm 0.001}$	-	-
w/o Generation	-	-	$0.457_{\pm 0.000}$	$0.569_{\pm 0.000}$	-	-	$0.421_{\pm 0.000}$	$0.505_{\pm 0.000}$

In this section, we investigate the effectiveness of fine-tuning ISIMUD on ConTSG-Bench (Lan et al., 2026), whose semantics are domain-specific and exhibit a clear gap from the domain-agnostic semantics used during pretraining. We fine-tune ISIMUD on a subset of the training data from each dataset for only one epoch, and compare its performance with baseline methods trained on the full datasets. As shown in Fig. 4, we report the average ranking of different methods from two perspectives: generation fidelity and semantic consistency. The results address **RQ3**, showing that *even with low-cost fine-tuning on limited data,* ISIMUD *outperforms existing full-shot methods while also demonstrating greater stability.* Moreover, its generative performance improves steadily as the number of training samples increases. We attribute this strong adaptability to large-scale pretraining on domain-agnostic semantics, which enables ISIMUD to capture shared knowledge across domains and efficiently transfer to downstream tasks. More detailed evaluation results are provided in Appendix G.

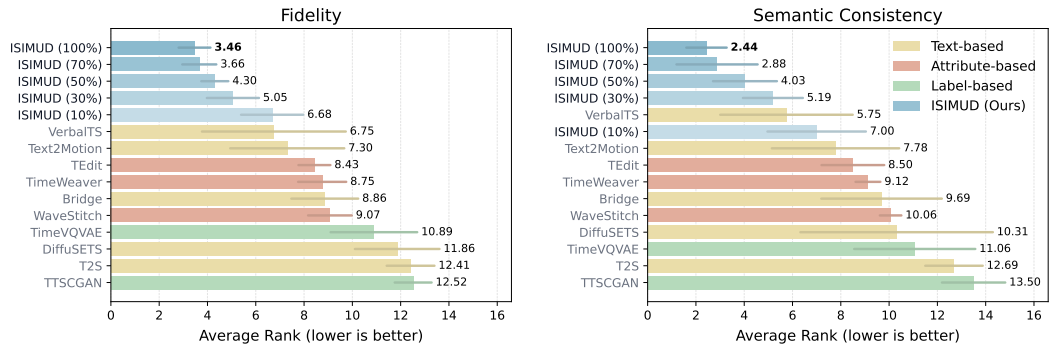

Figure 4: Few-shot generation evaluation on ConTSG-Bench. We consider both the fidelity (left) and semantic consistency (right). A lower average rank indicates better performance.

4.6 Question Answering

In this section, we extend ISIMUD to the question answering task and evaluate it on the ChatTime benchmark (Wang et al., 2025). This benchmark requires models to understand key features of the time series, including trend, seasonality, volatility, and outliers, for accurate answers. As shown in Fig. 5, ISIMUD achieves the best performance across all four features. These results address **RQ3**, demonstrating that ISIMUD *can efficiently adapt to question answering task with minimal additional cost.*

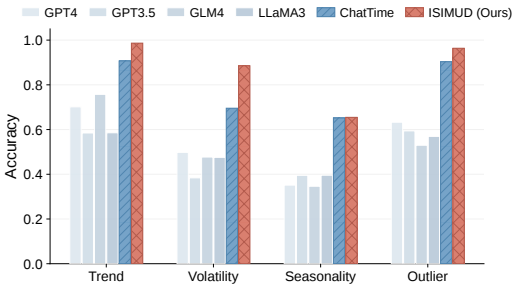

Figure 5: Finetune time series question answering on ChatTime benchmark. Higher accuracy indicates better time-series understanding.

4.7 Captioning Helps Forecasting

In this section, we validate the practical utility of time series captioning. Specifically, we compare the forecasting performance of large language models (LLMs) with and without the captioning text. For comprehensive evaluation, we consider both open-source LLM: Qwen3 (Yang et al., 2025), and closed-source models: GPT4o (Hurst et al., 2024) and Claude4.5 (Anthropic, 2025). As shown in Fig. 6, *incorporating the captioning text generated by* ISIMUD *consistently improves the forecasting performance across all models.* We attribute the improvement that captioning provides an additional semantic perspective for time series patterns, enhancing the model to capture underlying dynamics and improving forecasting.

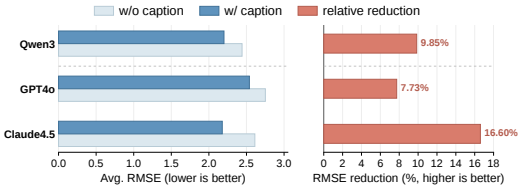

Figure 6: Forecasting performance of different large language models with or without the captioning text generated by ISIMUD.

5 Conclusion

In this work, we unify the dual tasks of time series generation and captioning, which are dual problems and both require cross-modal semantic alignment. However, existing works remain constrained by task specificity and dataset specificity. The semantics of time series vary significantly across domains, leading to scarcity of high-quality paired data, while the modality gap between text and time series further hinders the development of a unified method. To address these challenges, we construct BOOK-OF-ISIMUD, a large-scale multimodal dataset with semantics applicable across multiple domains, and propose ISIMUD within a single diffusion-based architecture, unified masked-conditional training, and semi-autoregressive inference paradigm. Extensive experiments show that ISIMUD performs well on both in-domain and zero-shot generation and captioning, and adapts efficiently to downstream tasks. We further demonstrate the benefits of unified training and semi-autoregressive inference in both in-domain and zero-shot settings. We believe this work provides a promising direction for unified multimodal foundation models in time series.

THE USAGE OF GENERATIVE AI

In this work, we primarily leverage generative AI for data synthesis. As discussed in Appendix D, we design an agent-collaborative textual annotation strategy to construct large-scale, high-quality paired time series–text datasets. This initiative aims to provide sufficient and reliable training data for the development of time series foundation models. We further assess the quality of the synthesized data through both automatic and human evaluations in Appendix D.2, demonstrating the effectiveness of our annotation strategy.

ETHICS STATEMENT

Our method offers several potential positive societal impacts: (i) enabling the generation of richer multimodal time series data across diverse domains, (ii) reducing the risk of personal privacy leakage through the use of synthetic data, (iii) helping human more efficiently and effectively understand time series, and (iv) supporting data-driven decision-making across a wide range of applications.

While we do not anticipate any direct negative societal consequences from this work, it is important to acknowledge that, like all generative technologies, it may be subject to misuse and therefore warrants careful oversight and responsible deployment.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide comprehensive details on data construction, model architecture, training, and evaluation throughout the paper. Specifically, the sources of the time-series data are described in Appendix C.1, the text annotation methods are detailed in Appendix D, and the model configurations, together with the training and inference parameters, are provided in Appendix F. Upon acceptance, we will release the code and datasets to further facilitate reproducibility and support future research.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Anthropic. System card: Claude sonnet 4.5, 2025. `https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf`.

Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=tyEyYT267x`.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307: 72–77, 2018.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

Yuan Gao, Haokun Chen, Xiang Wang, Zhicai Wang, Xue Wang, Jinyang Gao, and Bolin Ding. Diffsformer: A diffusion transformer on stock factor augmentation. *arXiv preprint arXiv:2402.06656*, 2024.

Yunfeng Ge, Jiawei Li, Yiji Zhao, Haomin Wen, Zhao Li, Meikang Qiu, Hongyan Li, Ming Jin, and Shirui Pan. T2s: high-resolution time series generation with text-to-series diffusion models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 5208–5216, 2025.

Shuqi Gu, Chuyue Li, Baoyu Jing, and Kan Ren. Verbalts: Generating time series from texts. *International Conference on Machine Learning*, 2025.

Shuqi Gu, Yongxiang Zhao, Baoyu Jing, and Kan Ren. What if tomorrow is the world cup final? counterfactual time series forecasting with textual conditions. In *International Conference on Machine Learning (ICML)*, 2026.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=kOIclg7muL.

Tong Guan, Sheng Pan, Johan Barthelemy, Zhao Li, Yujun Cai, Cesare Alippi, Ming Jin, and Shirui Pan. Timeomni-vl: Unified models for time series understanding and generation. *arXiv preprint arXiv:2602.17149*, 2026b.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022.

Huan He, Shifan Zhao, Yuanzhe Xi, and Joyce C Ho. Meddiff: Generating electronic health records using accelerated denoising diffusion model. *arXiv preprint arXiv:2302.04355*, 2023.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Daniel Jarrett, Jinsung Yoon, Ioana Bica, Zhaozhi Qian, Ari Ercole, and Mihaela van der Schaar. Clairvoyance: A pipeline toolkit for medical time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xnC8YwKUE3k.

Baoyu Jing, Hanghang Tong, and Yada Zhu. Network of tensor time series. In *Proceedings of the Web Conference 2021*, pp. 2425–2437, 2021.

Baoyu Jing, Shuqi Gu, Tianyu Chen, Zhiyu Yang, Dongsheng Li, Jingrui He, and Kan Ren. Towards editing time series. *Advances in Neural Information Processing Systems*, 2024a.

Baoyu Jing, Yansen Wang, Guoxin Sui, Jing Hong, Jingrui He, Yuqing Yang, Dongsheng Li, and Kan Ren. Automated contrastive learning strategy search for time series. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4612–4620, 2024b.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 36:28341–28364, 2023.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025.

Yongfan Lai, Jiabo Chen, Qinghao Zhao, Deyun Zhang, Yue Wang, Shijia Geng, Hongyan Li, and Shenda Hong. Diffusets: 12-lead ecg generation conditioned on clinical text reports and patient-specific information. *Patterns*, 2025.

Shaocheng Lan, Shuqi Gu, Zhangzhi Xiong, and Kan Ren. Contsg-bench: A unified benchmark for conditional time series generation. *arXiv preprint arXiv:2603.04767*, 2026.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A Xu, Winnie Chow, Martin Maritsch, Robert Jakob, Ning Wang, Juncheng Liu, Aradhana Verma, Brian Han, Daniel Seung Kim, Henry Chubb, Scott R. Ceresnak, Aydin Zahedivash, Alexander T Sandhu, Fatima Rodriguez, Daniel McDuff, Elgar Fleisch, Oliver Oppers Aalami, Filipe Barata, and Paul Schmiedmayer. OpenTSLM: Time-series language models for reasoning over multivariate medical text- and time-series data. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FHDRzhKm0f.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In *International Conference on Artificial Intelligence and Statistics*, pp. 7665–7693. PMLR, 2023.

Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. Timecap: Learning to contextualize, augment, and predict time series events with large language model agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18082–18090, 2025.

Hao Li, Yu-Hao Huang, Chang Xu, Viktor Schlegel, Renhe Jiang, Riza Batista-Navarro, Goran Nenadic, and Jiang Bian. Bridge: Bootstrapping text to control time-series generation via multi-agent iterative optimization and diffusion modeling. In *International Conference on Machine Learning*, pp. 34742–34773. PMLR, 2025a.

Xiaomin Li, Anne Hee Hiong Ngu, and Vangelis Metsis. Tts-cgan: A transformer time-series conditional gan for biosignal data augmentation. *arXiv preprint arXiv:2206.13676*, 2022.

Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2779–2790, 2025b.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR) 2023*, 2023. URL https://iclr.cc/virtual/2023/poster/11309.

Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep P. Chinchali. Time weaver: A conditional time series generation model. In *ICML*, 2024. URL https://openreview.net/forum?id=WpKDeixmFr.

Medhasweta Sen, Zachary Gottesman, Jiaxing Qiu, C Bayan Bruss, Nam Nguyen, and Tom Hartvigsen. Bedtime: A unified benchmark for automatically describing time series. *arXiv preprint arXiv:2509.05215*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems 37*. Neural Information Processing Systems Foundation, Inc., 2024. doi: 10.52202/079017-3277.

Mohamed Trabelsi, Aidan Boyd, Jin Cao, and Huseyin Uzunalioglu. Time series language model for descriptive caption generation. *Engineering Applications of Artificial Intelligence*, 162:112673, 2025.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):1–15, 2020.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, pp. 53140–53164. PMLR, 2024.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *Proceedings of the VLDB Endowment*, 18(8):2385–2398, 2025.

Zhijian Xu, Yuxuan Bian, Jianyuan Zhong, Xiangyu Wen, and Qiang Xu. Beyond trend and periodicity: Guiding time series forecasting with textual cues. *arXiv preprint arXiv:2405.13522v1*, 2024. URL https://arxiv.org/abs/2405.13522v1.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, and Kaifu Zhang. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2026. URL https://arxiv.org/abs/2505.02567.

Luca Zhou, Pratham Yashwante, Marshall Fisher, Alessio Sampieri, Zihao Zhou, Fabio Galasso, and Rose Yu. Cats-bench: Can language models describe numeric time series? *arXiv preprint arXiv:2509.20823*, 2025.

A LIMITATIONS

Despite its notable strengths, ISIMUD still has several limitations that may affect its broader applicability. Currently, the framework is primarily designed for time series generation and captioning, and requires additional fine-tuning to adapt to downstream tasks such as question answering. In addition, the model lacks explicit reasoning capabilities, which limits its effectiveness in more complex understanding scenarios.

Looking ahead, promising directions for improvement include extending the framework to natively support a wider range of tasks and incorporating stronger reasoning abilities. Such advancements could broaden the practical scope of ISIMUD and enhance its performance in dynamic, data-intensive environments.

B PRELIMINARY STUDY: CONTINUOUS AND DISCRETE DIFFUSION

Continuous diffusion. Flow matching (Lipman et al., 2023) is a generative framework that learns a transport map between a simple prior source distribution $p_0(\mathbf{x})$ and a complex target distribution $p_1(\mathbf{x}|\mathbf{c})$ conditioned on a variable $\mathbf{c}$, where $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional continuous variable. This transport is governed by a Probability Flow Ordinary Differential Equation (ODE):

$$\frac{d}{dt}\psi_t(\mathbf{x}) = v_t(\psi_t(\mathbf{x})|\mathbf{c}) \text{ and } \psi_0(\mathbf{x}) = \mathbf{x}, \tag{3}$$

where $t \in [0, 1]$, v_t is a time-dependent vector field determining a flow ψ_t. Directly regressing the marginal vector field is intractable because the true density p_t is unknown. To circumvent this, we utilize Conditional Flow Matching. For a given pair of samples $(\mathbf{x}_0, \mathbf{x}_1)$, we define a conditional probability path $\mathbf{x}_t \sim p_t$ and a corresponding conditional vector field $v_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1)$:

$$\begin{aligned} \mathbf{x}_t &= \psi_t(\mathbf{x}_0, \mathbf{x}_1) = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \\ v_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x_1}) &= \frac{d}{dt}\psi_t(\mathbf{x}_0, \mathbf{x}_1) = \mathbf{x}_1 - \mathbf{x}_0. \end{aligned} \tag{4}$$

The model $v_t(\mathbf{x}_t, t, \mathbf{c})$ is trained to approximate the target vector field v_t:

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]} \left\| v_\theta(\mathbf{x}_t, t, \mathbf{c}) - (\mathbf{x}_1 - \mathbf{x}_0) \right\|^2. \tag{5}$$

During inference, starting from a random noise $\mathbf{x}_0 \sim \mathcal{N}(0, I)$, we generate data by integrating the learned vector field v_θ conditioned on $\mathbf{c}$. The final state $\mathbf{x}_1$ is obtained using an Euler ODE solver:

$$\mathbf{x}_1 = \mathbf{x}_0 + \int_0^1 v_\theta(\mathbf{x}_t, t, \mathbf{c})dt \tag{6}$$

Discrete Diffusion. The masked diffusion model (Shi et al., 2024) is a generative model that learns to reverse a corruption process. Given a discrete data sequence $\mathbf{x} \in \{1, \ldots, K\}^L$ with K categories and length L, we introduce a special absorption state $\mathbf{m}$, representing the [MASK] token, and define the forward corruption process $q(\mathbf{x}_t \mid \mathbf{x}_0)$, which progressively replaces tokens with $\mathbf{m}$. The transition probability is expressed as:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \text{Cat}(\mathbf{x}_t; \alpha_t\mathbf{x}_0 + (1-\alpha_t)\mathbf{m}), \tag{7}$$

where Cat denotes the categorical distribution, and α_t controls the extent of corruption. The model $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ is trained to reverse this corruption process by minimizing the negative evidence lower bound objective (NELBO):

$$\mathcal{L}_{\text{MDM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]} \left[-\frac{1}{t} \sum_{i=1}^{L} \mathbf{1}[\mathbf{x}_t^i = \mathbf{m}] \log p_\theta(\mathbf{x}_0^i|\mathbf{x}_t) \right], \tag{8}$$

where $\mathbf{1}[\mathbf{x}_t^i = \mathbf{m}]$ is an indicator function ensuring that the loss is only computed for the tokens replaced by [MASK]. The factor $\frac{1}{t}$ acts as a time-dependent weighting factor, which is derived from the diffusion dynamics under the linear noise schedule parameterized by α_t.

C Dataset Details

C.1 Book-of-Isimud

Book-of-Isimud is a paired text–time-series corpus for time series captioning and text-conditioned generation. It is compiled from 16 real-world datasets drawn from Chronos (Ansari et al., 2024) and Moirai (Woo et al., 2024), spanning five domains: Energy, Transport, Weather, Health, and Finance (Business & Economy), and contains approximately 100M time steps.

Table 4 reports the composition of Book-of-Isimud, including the number of raw time steps, the sample counts in each split, and the source benchmark of each constituent dataset. The corpus is designed to provide broad coverage of cross-domain temporal phenomena while maintaining a unified annotation scheme. Unless otherwise specified, the datasets in Table 4 are used for training and in-domain evaluation. Following the protocol in Sec. 4, weatherbench and fox are held out from training and used exclusively for zero-shot evaluation.

Table 4: Statistics of datasets in Book-of-Isimud. The datasets are grouped by domain. Weatherbench and fox are used for zero-shot evaluation, while the remaining datasets are used for in-domain training and evaluation.

Domain	Dataset	Time Steps	Train Samples	Valid Samples	Test Samples	Source
Energy	bear	1,482,312	48,984	6,084	6,240	Moirai
	fox†	2,324,568	14,440	1,720	1,990	Moirai
	rat	4,728,288	29,365	3,493	4,053	Moirai
	ercot	1,238,976	41,264	5,152	5,168	Chronos
	hog	421,056	13,920	1,728	1,776	Moirai
	electricity	8,443,584	52,644	6,420	6,741	Chronos
	wind_power	7,397,147	46,232	5,779	5,779	Moirai
Transport	city_bikes	38,687,004	241,417	29,931	30,643	Chronos
Weather	era5	33,808,320	211,216	26,402	26,488	Moirai
	weatherbench†	640,000	4,000	500	500	Chronos
Health	covid_deaths	56,392	18,088	2,128	2,394	Chronos
	project_tycho	1,372,991	7,414	1,145	1,628	Moirai
Finance	exchange_rate	60,704	3,976	496	504	Chronos
	favorita_transactions	84,290	8,851	1,092	1,146	Moirai
	fred_md	77,896	51,360	6,420	6,527	Chronos
	tourism_monthly	109,189	50,041	6,125	6,668	Chronos
Total		**100,932,717**				

† Zero-shot datasets. They are excluded from in-domain training and used only for zero-shot generation and captioning evaluation.

C.2 Evaluation Settings

We evaluate under two settings: in-domain and zero-shot. In the in-domain setting, the model is trained on the training split and evaluated on the test split of the same dataset. In the zero-shot setting, the model is evaluated on datasets that are entirely held out from training, testing its generalization ability to unseen datasets.

In-domain evaluation. For in-domain evaluation, each dataset is split into training, validation, and test subsets with a ratio of 8:1:1. We report results on five representative datasets: era5 (Weather), ercot (Energy), city_bikes (Transport), covid_deaths (Health), and favorita_transactions (Finance). Together, these datasets cover all five domains in Book-of-Isimud.

Zero-shot evaluation. For zero-shot evaluation, we use weatherbench (Weather) and fox (Energy) as held-out datasets. These datasets are excluded from training and used exclusively to evaluate the zero-shot generation and captioning capabilities of Isimud.

D ISIMUD TEXTUAL ANNOTATION DETAILS

This section details the construction of the textual annotations in BOOK-OF-ISIMUD. Each time-series sample is paired with two complementary textual fields: a Domain-Agnostic morphology caption and a Domain-Specific context caption. The former describes the observable temporal morphology of the sequence, while the latter provides verified dataset- and channel-level semantic context. We first describe how these two textual fields are constructed, then present the prompt templates used by the annotation agents, and finally evaluate the faithfulness and overall quality of the resulting descriptions.

D.1 AGENT-COLLABORATIVE TEXTUAL ANNOTATION

The two textual fields serve different purposes. The **Domain-Agnostic morphology caption** describes the shape of the time series, including trend, seasonality, abrupt changes, volatility, local extrema, and repeating patterns, without using domain terminology, units, labels, or causal explanations. In contrast, the **Domain-Specific context caption** describes the corresponding domain and dataset information, such as whether the sequence represents energy demand, traffic flow, weather variables, physiological measurements, or business indicators. By separating these two fields, we can preserve faithful morphology descriptions while also providing domain semantics when such information is available.

The morphology annotation process is agent-collaborative and is implemented as a two-stage pipeline instantiated with Qwen3 (Yang et al., 2025). As described in Sec. 3.3, the **Describer** is responsible for generating diverse and fluent candidate descriptions from raw numerical sequences, while the **Refiner** is responsible for improving factual accuracy by verifying the draft against statistical evidence extracted from the time series.

Specifically, to obtain the domain-agnostic caption, the Describer is prompted with the raw numerical sequence and asked to summarize only its morphology in general natural language. The Refiner then revises this draft using tool-derived statistical evidence. The resulting refined paragraph is used as the final domain-agnostic morphology caption.

To obtain the domain-specific context caption, we combine agent-assisted web search with human confirmation. The web-search agent is used only to collect dataset-level background information, such as the application domains and variable meanings. The retrieved information is then manually checked and confirmed before being used for annotation.

D.1.1 THE DESCRIBER

Given a univariate sequence serialized in temporal order, the Describer D outputs one concise paragraph summarizing the curve morphology. It focuses on three levels of structure: the global trajectory, sustained regime-level changes, and salient local events such as peaks, troughs, abrupt shifts, and unusually flat intervals. When clearly supported, it also identifies the temporal location of the global maximum and minimum. Throughout, D remains domain-agnostic, avoids precise numeric reporting, and explicitly marks uncertainty when the pattern is ambiguous.

Prompt: Describer

Role
You are a domain-agnostic describer of time-series morphology.
Input
A univariate time series given as numeric values in chronological order.
Do not assume any domain, units, labels, or causes.
Task
Write one concise paragraph describing the visible morphology of the series.
Cover the global trajectory, sustained regime changes, salient local events, extreme placement, variability, and repeating structure only when clearly supported.
Guidelines
Stay domain-agnostic.
Describe observable structure rather than interpretations or causes.
Avoid exact numbers, timestamps, units, and metadata.
If the pattern is ambiguous, state the uncertainty rather than guessing.
Output Constraints
Output only the paragraph.
Do not produce lists, headings, or structured data.

D.1.2 THE REFINER

The Refiner R improves faithfulness by checking the draft against a compact evidence summary extracted with `tsfresh` (Christ et al., 2018). This summary is computed for the full series as well as early, middle, and late segments, and captures broad direction, extreme timing, fluctuation scale, short-term irregularity, curvature, sustained deviations, and support for repeating structure. To make evidence strength comparable across heterogeneous datasets, selected statistics are converted to empirical percentiles computed over the corresponding source dataset.

Conditioned on the draft and the evidence summary, R performs a verification-driven editing pass: supported statements are retained, contradicted claims are revised or removed, and new details are added only when multiple evidence signals agree. The refined paragraph is used as the final textual annotation.

Prompt: Refiner

Role
You are a domain-agnostic editor of time-series morphology captions.
Inputs
(1) A draft morphology caption produced from the raw sequence.
(2) A structured evidence summary extracted from the time series.
Task
Produce one refined paragraph that is faithful, concise, and evidence-grounded.
Evidence Usage Guidelines
The evidence summary covers trend and its stability, extreme timing, fluctuation scale, short-term irregularity, curvature, sustained deviations, and support for repeating structure.
It provides these signals both globally and across coarse temporal segments.
Editing Rules
Retain supported statements.
Revise or remove contradicted claims.
Add missing morphology details only when multiple evidence signals agree.
Do not mention feature names, thresholds, or raw statistics.
Output Constraints
Output only the refined paragraph.
Do not produce lists, headings, or structured data.
Avoid exact numbers, units, and causal explanations.

D.2 Quality Evaluation of the Textual Annotation

We evaluate the quality of the constructed morphology captions from two complementary perspectives. First, we measure cross-modal consistency using CTTP-based bidirectional retrieval accuracy. Second, we conduct a blind forced-choice human evaluation comparing the initial captions produced by the Describer against the final captions after refinement.

Automatic Evaluation via CTTP-Based Retrieval Accuracy. To quantify whether refinement improves cross-modal alignment, we evaluate the draft and refined captions using the same CTTP retrieval protocol Gu et al. (2025). Specifically, each retrieval query is evaluated as a 32-way classification problem over 32 candidates in the opposite modality, and we report time-series-to-text retrieval accuracy (TS→Text Acc.) and text-to-time-series retrieval accuracy (Text→TS Acc.), together with their arithmetic mean. As shown in Table 5, the final captions substantially improve bidirectional retrieval accuracy over the initial captions (significantly higher than the chance accuracy 3.1%), indicating stronger semantic alignment between the textual annotations and the underlying time series.

Table 5: CTTP-based bidirectional retrieval accuracy of the initial and final captions. Arrow ↑ indicates that higher is better.

Text Annotation	**TS→Text Acc. ↑**	**Text→TS Acc. ↑**	**Avg. ↑**
Initial Text	0.3470	0.3552	0.3511
Final Text	**0.4738**	**0.4751**	**0.4744**

Human Evaluation. We further conduct a blind forced-choice human evaluation without a tie option. Four evaluators with relevant domain expertise compare draft captions against their refined counterparts on instances sampled from all five domains. For each pair, the draft and refined captions are anonymized as two candidate descriptions and presented in randomized order, so evaluators are blind to the method identity of each caption. They judge each pair along three criteria: overall quality, faithfulness to the time series, and fewer unsupported claims. Table 6 summarizes the win rates. Across all three criteria, final captions are preferred more often than draft captions.

Table 6: Win rates in blind forced-choice human evaluation of the initial and final captions. Arrow ↑ indicates that higher is better.

Criterion	**Initial Text Win Rate (%) ↑**	**Final Text Win Rate (%) ↑**
Overall	42.7	**57.3**
Faithfulness	45.3	**54.7**
Fewer Unsupported Claims	36.0	**64.0**

E Evaluation Metrics

In this section, we provide the definitions of the evaluation metrics used in our experiments. Given a pair of time series and textual descriptions $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^L$ denotes a time series and y_i denotes its corresponding text, the generation task produces $\hat{x}_i$ conditioned on y_i, while the captioning task produces $\hat{y}_i$ conditioned on x_i. We use a fixed pretrained CTTP (Gu et al., 2025) evaluator, consisting of time-series and text encoders $\phi_{\text{ts}}(\cdot)$ and $\phi_{\text{text}}(\cdot)$, to embed time series and texts into a shared cross-modal representation space, respectively. Unless otherwise specified, all CTTP-based metrics in this paper use this same checkpoint.

E.1 J-FTSD

For time series generation, we use Joint Frechet Time Series Distance (J-FTSD) to evaluate the joint distributional discrepancy between real text-time series pairs and generated text-time series pairs.

Following prior conditional time series generation benchmarks (Lan et al., 2026; Gu et al., 2025), we construct joint embeddings by concatenating the time series embedding and the text embedding:

$$z_i^r = \phi_{\text{ts}}(x_i) \oplus \phi_{\text{text}}(y_i), \qquad z_i^g = \phi_{\text{ts}}(\hat{x}_i) \oplus \phi_{\text{text}}(y_i), \tag{9}$$

where $\oplus$ denotes vector concatenation, z_i^r is the real joint embedding, and z_i^g is the generated joint embedding.

Let (μ_r, Σ_r) and (μ_g, Σ_g) denote the empirical mean and covariance of the real and generated joint embeddings, respectively. J-FTSD is then defined as:

$$\text{J-FTSD} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right). \tag{10}$$

A lower J-FTSD indicates that the generated time series better match the real text-time series joint distribution.

E.2 CTTP Score

We use Contrastive Text-Time Series Pretraining (CTTP) model score to measure cross modal semantic consistency between time series and text (Gu et al., 2025). Different from unimodal similarity metrics, CTTP is always computed between one time-series representation and one text representation. We define the cosine similarity between two embeddings u and v as:

$$\text{sim}(u, v) = \frac{u^\top v}{\|u\|_2 \|v\|_2}. \tag{11}$$

For the generation task, CTTP measures the semantic alignment between the generated time series $\hat{x}_i$ and the ground-truth textual description y_i:

$$\text{CTTP}_{\text{gen}} = \frac{1}{N}\sum_{i=1}^{N} \text{sim}\left(\phi_{\text{ts}}(\hat{x}_i), \phi_{\text{text}}(y_i)\right). \tag{12}$$

For the captioning task, CTTP measures the semantic alignment between the real time series x_i and the generated textual description $\hat{y}_i$:

$$\text{CTTP}_{\text{cap}} = \frac{1}{N}\sum_{i=1}^{N} \text{sim}\left(\phi_{\text{ts}}(x_i), \phi_{\text{text}}(\hat{y}_i)\right). \tag{13}$$

A higher CTTP score indicates stronger cross-modal semantic consistency. In our experiments, CTTP_{gen} is reported for generation, and CTTP_{cap} is reported for captioning.

E.3 ROUGE-L

For the captioning task, we use ROUGE-L Lin (2004) to evaluate the textual overlap between the generated caption $\hat{y}_i$ and the reference caption y_i. ROUGE-L is based on the longest common subsequence (LCS) between two token sequences. For each sample, let $\text{LCS}(\hat{y}_i, y_i)$ denote the length of the longest common subsequence. The precision and recall are computed as:

$$P_i = \frac{\text{LCS}(\hat{y}_i, y_i)}{|\hat{y}_i|}, \qquad R_i = \frac{\text{LCS}(\hat{y}_i, y_i)}{|y_i|}. \tag{14}$$

We report the ROUGE-L score:

$$\text{ROUGE-L} = \frac{1}{N}\sum_{i=1}^{N} \frac{2P_iR_i}{P_i + R_i}. \tag{15}$$

A higher ROUGE-L score indicates better structural overlap with the reference caption.

E.4 Metrics for Few-Shot Conditional Generation

For few-shot conditional generation on ConTSG-Bench, we follow the official ranking-based evaluation protocol (Lan et al., 2026). All raw metrics and aggregate rankings are computed with the official benchmark code to ensure consistency with the reported baselines. The evaluation is summarized from two perspectives: generation fidelity and semantic consistency. Generation fidelity measures whether the generated time series resemble real time series, while semantic consistency measures whether the generated time series follow the given conditions.

Following ConTSG-Bench, the generation fidelity group includes both embedding-based and statistical metrics. Specifically, we consider Frechet Inception Distance (FID) in the CTTP time-series embedding space, precision and recall in the same embedding space, marginal distribution difference (MDD), auto-correlation difference (ACD), skewness difference (SD), and kurtosis difference (KD). These metrics evaluate whether the generated samples match the marginal distribution and temporal statistics of real time series.

The semantic consistency group evaluates whether the generated time series are aligned with the input conditions. We include CTTP score, Joint Frechet Time Series Distance (J-FTSD), joint precision, and joint recall. For these joint-space metrics, each sample is represented by the concatenation of its time-series embedding and condition embedding. In our setting, the condition is represented by the corresponding textual description.

Let $\mathcal{D}_{\mathrm{fs}}$ denote the set of few-shot evaluation datasets, and let $\mathcal{M}_{\mathrm{fid}}$ and $\mathcal{M}_{\mathrm{sem}}$ denote the metric sets for fidelity and semantic consistency, respectively:

$$\mathcal{M}_{\mathrm{fid}} = \{\mathrm{FID}, \mathrm{Precision}, \mathrm{Recall}, \mathrm{MDD}, \mathrm{ACD}, \mathrm{SD}, \mathrm{KD}\}, \tag{16}$$

$$\mathcal{M}_{\mathrm{sem}} = \{\mathrm{CTTP}, \text{J-FTSD}, \mathrm{JointPrecision}, \mathrm{JointRecall}\}. \tag{17}$$

For each dataset d, metric m, and method a, we first compute the raw metric score $s_{d,m}(a)$. Since different metrics have different directions and numerical ranges, we rank methods according to the proper direction of each metric, where rank 1 indicates the best performance. For distance-based metrics, lower values are better, while for similarity-based metrics, higher values are better.

The average rank for a metric group $\mathcal{M}$ is computed as:

$$\mathrm{AvgRank}_{\mathcal{M}}(a) = \frac{1}{|\mathcal{D}_{\mathrm{fs}}||\mathcal{M}|} \sum_{d \in \mathcal{D}_{\mathrm{fs}}} \sum_{m \in \mathcal{M}} \mathrm{rank}_{d,m}(a). \tag{18}$$

We report $\mathrm{AvgRank}_{\mathcal{M}_{\mathrm{fid}}}$ for generation fidelity and $\mathrm{AvgRank}_{\mathcal{M}_{\mathrm{sem}}}$ for semantic consistency. A lower average rank indicates better overall performance.

E.5 Metrics for Question Answering

For time series question answering, we report accuracy (Acc). Given N test questions, let $\hat{a}_i$ be the predicted answer and a_i be the ground-truth answer. The accuracy is defined as:

$$\mathrm{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\hat{a}_i = a_i], \tag{19}$$

where $\mathbb{I}[\cdot]$ is the indicator function. When questions are grouped by semantic type, such as trend, seasonality, volatility, and outlier, we additionally report category-level accuracy:

$$\mathrm{Acc}_c = \frac{1}{|\mathcal{D}_c|} \sum_{i \in \mathcal{D}_c} \mathbb{I}[\hat{a}_i = a_i], \tag{20}$$

where $\mathcal{D}_c$ denotes the subset of questions belonging to category c.

E.6 Metrics for Caption-Augmented Forecasting

For the caption-augmented LLM forecasting setting, we report root mean squared error (RMSE). After parsing the LLM output, both forecasts and targets are evaluated on the original value scale

rather than on a normalized scale. Let $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,H})$ denote the ground-truth future sequence for the i-th test instance, and let $\hat{\mathbf{u}}_i = (\hat{u}_{i,1}, \ldots, \hat{u}_{i,H})$ denote the corresponding forecast over a prediction horizon of length H. RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{NH} \sum_{i=1}^{N} \sum_{t=1}^{H} (u_{i,t} - \hat{u}_{i,t})^2}. \tag{21}$$

A lower RMSE indicates better forecasting performance.

F Model Implementation and Experimental Settings

In this section, we provide the implementation details and experimental settings. We first summarize the model configuration, including the shared backbone, modality-specific encoders and decoders, and key architectural hyperparameters. We then describe the training and inference configurations, evaluation protocols, downstream adaptation settings, and baseline implementation details.

F.1 Model Configuration

ISIMUD is implemented as a unified multimodal diffusion model for time series captioning and text-conditioned time series generation. The model uses a pretrained diffusion language model (Cheng et al., 2025) as the shared backbone for both textual and time series embeddings. The time series modality is interfaced through lightweight patch-based encoder-decoder modules, which are trained from scratch, with model dimension 2048 and patch length 8. For captioning, next-block prediction uses a block size of 32. For generation, next-scale prediction operates over three scales, progressing from coarse to fine resolution.

For next-scale prediction, we construct multi-scale time series using moving average filters. Given a time series $\mathbf{x} = [x_1, \ldots, x_L] \in \mathbb{R}^L$, where x_t denotes the scalar observation at temporal position t, we generate L_B^{ts} scale-specific sequences $\{\mathbf{x}^{(i)}\}_{i=1}^{L_B^{\mathrm{ts}}}$ as

$$\mathbf{x}^{(i)} = \mathrm{MA}(\mathbf{x}, w_i), \qquad w_i = k^{L_B^{\mathrm{ts}} - i}, \qquad i = 1, \ldots, L_B^{\mathrm{ts}}, \tag{22}$$

where k is the base kernel size and w_i is the moving-average window size at scale i. In our implementation, $L_B^{\mathrm{ts}} = 3$, corresponding to *coarse, medium*, and *fine* scales. The last scale uses $w_{L_B^{\mathrm{ts}}} = 1$, which preserves the original time-series resolution.

The moving average is implemented as one-dimensional average pooling over a padded sequence. Let

$$\widetilde{\mathbf{x}} = \mathrm{Pad}_{w-1}(\mathbf{x}) \tag{23}$$

denote the padded sequence obtained by padding $\mathbf{x}$ with $w-1$ positions before the first observation. For each temporal position $t \in \{1, \ldots, L\}$, the smoothed value is computed as

$$[\mathrm{MA}(\mathbf{x}, w)]_t = \frac{1}{w} \sum_{\ell=0}^{w-1} \widetilde{x}_{t+\ell}. \tag{24}$$

This corresponds to applying average pooling with kernel size w and stride 1 to the padded sequence. Here, t denotes the temporal position in the output sequence, and ℓ indexes the positions inside the moving-average window. The padding keeps the output sequence length equal to the input length L.

The unified architecture is shown in Fig. 7. Text and time series are first mapped into a shared latent space by modality-specific encoders. The resulting latent blocks are then processed by a shared Transformer backbone, which consists of stacked self-attention and feed-forward layers. Finally, task-specific decoders map the shared hidden states back to either text tokens or time-series patches. The right side of Fig. 7 illustrates the task-specific attention masks, with unified inter-block (scale) casual attention and intra-block (scale) full attention. For captioning, the model predicts text blocks conditioned on the time-series representation and previously generated text blocks. For generation, the model predicts time-series scales conditioned on the textual input and previously generated coarser-scale time series. This design allows captioning and generation to share the same backbone while preserving different autoregressive dependency structures for the two tasks.

F.2 Training and Inference Configuration

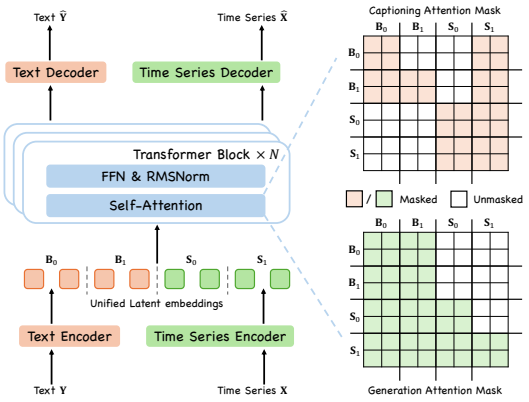

Figure 7: The model architecture of ISIMUD.

Pretraining. ISIMUD is pretrained on BOOK-OF-ISIMUD using paired time series and textual descriptions. For each paired sample, the model is trained in both directions: text-conditioned time series generation and time series captioning. The generation objective models the continuous time-series modality with flow matching, while the captioning objective models the discrete text modality with masked diffusion. During pretraining, we fully fine-tune the shared diffusion language model backbone rather than keeping it frozen, jointly optimizing it with the time-series encoder and decoder.

We train ISIMUD for 20 epochs on 8 NVIDIA A100 GPUs using distributed mixed-precision training, corresponding to approximately 384 GPU hours in total. We optimize the model with Adam using a learning rate of 1×10^{-4}, weight decay 1×10^{-6}, and a batch size of 16 per GPU. The generation and captioning losses are weighted by $\lambda_{\text{gen}} : \lambda_{\text{cap}} = 2 : 1$. Unless otherwise specified, the same pretrained checkpoint is used for in-domain evaluation, zero-shot evaluation, and downstream adaptation.

Inference. For text-conditioned time series generation, ISIMUD follows the next-scale prediction paradigm. Starting from the coarsest scale, the model progressively generates time series from coarse to fine resolutions. Within each scale, samples are generated by integrating the learned vector field with an Euler solver. Unless otherwise specified, we use 20 sampling steps for each textual condition. For time series captioning, ISIMUD follows the next-block prediction paradigm. The model generates text blocks autoregressively, while tokens within each block are denoised in parallel through masked diffusion. Caption generation stops when the end-of-sequence token is produced or when the maximum generation length is reached. For captioning, we use 8 diffusion denoising steps per text block, with a block size of 32 tokens. We ran the experiments using three random seeds and report the mean and standard deviation.

F.3 EVALUATION AND DOWNSTREAM ADAPTATION PROTOCOLS

In-domain and zero-shot evaluation. For both in-domain and zero-shot evaluation, we directly apply the pretrained model without additional task-specific fine-tuning. In-domain results are reported on the test split of datasets whose training data are included in BOOK-OF-ISIMUD, whereas zero-shot results are reported on held-out datasets excluded from pretraining. Generation and captioning use the default inference hyperparameters described above.

Few-shot conditional generation. For few-shot conditional generation, we use ConTSG-Bench (Lan et al., 2026), which evaluates conditional time-series generation under target conditions that are more dataset-specific. We initialize ISIMUD from the same pretrained checkpoint used for in-domain and zero-shot evaluation, and fine-tune it independently on four ConTSG-Bench datasets: the synthetic dataset Synth, the two semi-synthetic datasets ETTm1 and Istanbul Traffic, and the real-world dataset Weather Conceptual. These datasets cover different data sources and condition semantics, allowing us to examine whether the pretrained model can transfer beyond its original semantic regime. For multivariate datasets, we reinitialize only the patch-based time-series encoder and decoder and train them from scratch during adaptation to accommodate dataset-specific channel configurations, while keeping the remaining parameters initialized from the pretrained checkpoint. We vary the amount of target training data to study data efficiency, including the 10%, 30%, 50%, 70%, and 100% settings. The 100% setting is included as an upper-bound adaptation reference, while the smaller ratios are used to characterize few-shot behavior. Unless otherwise specified, each adaptation run uses a single epoch of fine-tuning, and the model architecture and inference configuration are kept fixed across all data ratios.

For baselines, we follow the official ConTSG-Bench comparison protocol and include the methods reported on the corresponding benchmark splits. We use the benchmark-provided full-shot baseline results and compute our results with the official evaluation code to avoid implementation-dependent discrepancies. Evaluation follows the official ranking-based protocol described in Appendix E.4, and the complete per-dataset raw metrics are reported in Appendix G.

Question answering. For question answering, we evaluate on the ChatTime (Wang et al., 2025) QA benchmark, which tests time-series understanding through questions about trend, seasonality, volatility, and outliers. We follow the official ChatTime data splits and formulate each example as text generation conditioned on the input time series and the question prompt, without introducing any additional task-specific classifier or answer head. During adaptation, we fully fine-tune all model parameters initialized from the pretrained checkpoint, while keeping the same multimodal backbone as in the generation and captioning experiments. At inference time, the model directly generates the answer text, and we report category-level accuracy following the ChatTime evaluation protocol.

For baselines, we compare with the same models reported in the original ChatTime paper. The baseline numbers shown in Fig. 5 and Appendix G are taken directly from the original ChatTime paper rather than re-running the models. This choice avoids potential confounding factors from proprietary API updates, prompt-version differences, and unavailable implementation details, and keeps the comparison aligned with the published ChatTime benchmark. ChatTime is included as the task-specific time-series foundation model baseline, while the general-purpose LLM baselines provide a reference for how much time-series-specific pretraining and adaptation improve QA performance. We report accuracy on the four semantic question types used by ChatTime: trend, volatility, seasonality, and outlier.

Prompt: Caption-Augmented Forecasting

System Prompt
You are a careful time-series forecasting model.
Given the provided historical context, predict the next 64 observations. Return only a JSON array of exactly 64 numeric values in chronological order.
History-Only User Prompt
Historical time series with 128 observations:
{`history_values`}
Predict the next 64 observations. Return only the JSON array.
History-Caption User Prompt
Historical caption:
{`caption`}
Historical time series with 128 observations:
{`history_values`}
Predict the next 64 observations. Return only the JSON array.

G Full Results

G.1 Few-Shot Conditional Generation

We report the complete per-dataset results for the few-shot conditional generation experiments corresponding to Fig. 4. Best results are in bold and second-best results are underlined. Lower is better for ACD, SD, KD, MDD, FID, and J-FTSD; higher is better for Precision, Recall, Joint Precision, Joint Recall, and CTTP Score.

Table 7: Evaluation results for ETTm1 dataset.

Model	ACD	SD	KD	MDD	FID	J-FTSD	Precision	Recall	Joint Precision	Joint Recall	CTTP Score
Bridge	0.1069	**0.4925**	3.3521	0.0272	42.0076	42.6005	0.6698	0.4616	0.6089	0.4793	1.8156
DiffuSETS	0.1420	8.5331	258.7216	0.0424	56.4434	56.9916	0.6126	0.3859	0.5409	0.4806	1.9019
T2S	0.1150	1.1269	3.1332	0.0540	95.0026	95.6860	0.1115	0.3323	0.2886	0.2387	3.4514
TEdit	0.1692	0.7493	3.8786	0.0233	23.1472	23.8535	0.6089	0.5883	0.4828	0.5748	1.9972
Text2Motion	0.1114	0.8648	8.8360	0.0243	48.5253	48.9968	0.6105	0.2676	0.6579	0.4560	1.9365
TimeVQVAE	0.0530	1.9516	9.3044	0.0358	28.2658	28.7677	**0.7515**	0.3961	0.6195	0.6089	2.1357
TimeWeaver	0.0840	1.0313	4.1960	0.0284	48.9813	49.6649	0.6918	0.6074	0.4462	0.5810	2.6474
TTSCGAN	0.3025	1.4856	4.8669	0.0397	109.9494	110.6666	0.5916	0.0310	0.5872	0.2304	1.5059
VerbalTS	0.1580	1.7842	4.5994	0.0312	48.1654	48.7078	0.5001	0.4321	0.5983	0.4766	2.0389
WaveStitch	**0.0522**	0.8096	2.1095	0.0346	57.2002	57.7153	0.6116	0.4460	0.5980	0.4833	1.9481
ISIMUD (10%)	0.0856	1.6429	1.7148	0.0102	8.0783	8.7490	0.6482	0.8734	0.7392	0.9089	2.7439
ISIMUD (30%)	0.0699	1.3671	**1.5224**	**0.0094**	9.5650	10.0992	0.6276	0.8677	0.7898	0.9320	3.3141
ISIMUD (50%)	0.0583	1.5988	1.5724	0.0096	7.6647	8.1064	0.6644	0.9077	0.8596	0.9551	3.9162
ISIMUD (70%)	0.0563	1.5857	1.7465	0.0099	**7.0474**	**7.4320**	0.6669	**0.9233**	0.8702	**0.9607**	4.1296
ISIMUD (100%)	0.0618	1.7330	1.5292	0.0100	7.2800	7.6385	0.6812	0.9195	**0.8890**	0.9488	**4.2600**

Table 8: Evaluation results for Istanbul Traffic dataset.

Model	ACD	SD	KD	MDD	FID	J-FTSD	Precision	Recall	Joint Precision	Joint Recall	CTTP Score
Bridge	0.0294	0.4812	0.1500	0.0363	14.5301	15.0547	**0.9347**	0.2896	0.6885	0.5944	4.7030
DiffuSETS	0.1470	2.2346	12.9962	0.0408	28.6764	29.3117	0.5917	0.3923	0.6023	0.6127	4.7812
T2S	0.2451	0.3192	1.5178	0.0281	142.4714	143.6973	0.0000	0.1653	0.0052	0.0708	**5.4458**
TEdit	0.0310	0.2893	0.2992	0.0176	7.4086	8.1024	0.8277	0.8405	0.6223	0.7407	4.7114
Text2Motion	0.0573	0.1450	0.0779	**0.0102**	2.7859	3.1654	0.8548	0.7271	0.8559	0.8218	4.8171
TimeVQVAE	0.2015	0.3223	2.0618	0.0401	33.9157	34.7945	0.4548	0.1102	0.3241	0.3084	4.7204
TimeWeaver	0.0267	0.2055	0.5147	0.0200	8.5903	9.3096	0.6785	0.8306	0.6244	0.7515	4.7321
TTSCGAN	0.3399	0.1098	0.3162	0.0205	10.5484	11.4818	0.2352	0.1501	0.5300	0.6533	4.7343
VerbalTS	0.0236	0.1323	0.0921	0.0114	4.2518	4.6698	0.8212	0.8730	0.8438	0.8502	4.8572
WaveStitch	0.0215	0.4960	0.5819	0.0179	10.9867	11.7453	0.8430	0.8246	0.5964	0.7315	4.8229
ISIMUD (10%)	0.0601	0.0423	**0.0764**	0.0102	0.9942	1.4582	0.6899	0.8420	0.8595	**0.9705**	4.7682
ISIMUD (30%)	0.0115	0.0401	0.0881	0.0103	0.5876	1.0283	0.8206	0.8915	0.8902	0.9476	4.8189
ISIMUD (50%)	**0.0055**	0.0487	0.0987	0.0105	0.5990	1.0338	0.8457	0.9160	0.8975	0.9690	4.8171
ISIMUD (70%)	0.0147	0.0398	0.0968	0.0104	**0.4141**	**0.8386**	0.8667	0.9285	**0.9062**	0.9552	4.8514
ISIMUD (100%)	0.0348	**0.0360**	0.0887	0.0105	0.6194	1.0581	0.8567	**0.9351**	0.8971	0.9326	4.8592

Table 9: Evaluation results for Synth dataset.

Model	ACD	SD	KD	MDD	FID	J-FTSD	Precision	Recall	Joint Precision	Joint Recall	CTTP Score
Bridge	0.0314	0.0974	0.6417	0.0282	50.1814	55.1401	0.5906	0.0745	0.8228	0.4752	22.9919
DiffuSETS	0.0552	0.1404	0.8331	0.0203	41.5913	46.5630	0.5842	0.2054	0.8427	0.6600	24.6628
T2S	0.0489	0.4960	0.7833	0.0358	144.4827	156.3209	0.2330	0.0000	0.2862	0.0140	10.3251
TEdit	0.0658	0.1632	0.2876	0.0216	49.1590	59.9647	0.5448	0.1217	0.7358	0.3991	20.3756
Text2Motion	0.0821	0.0715	0.2046	0.0150	58.5985	66.6559	0.5509	0.0562	0.7030	0.3052	17.6067
TimeVQVAE	0.0800	0.0457	0.7758	0.0245	75.7424	83.9404	**0.7228**	0.0170	0.7329	0.2192	16.0165
TimeWeaver	0.0735	0.0469	0.5210	0.0207	55.4405	65.8629	0.6252	0.1102	0.7436	0.3437	19.0760
TTSCGAN	0.2849	0.1043	0.1352	0.0233	119.9663	132.8113	0.1867	0.0000	0.1907	0.0290	9.3852
VerbalTS	0.0601	0.0317	0.3739	0.0156	37.9622	**41.3352**	0.6828	0.2021	**0.9524**	0.7239	27.2469
WaveStitch	0.0491	0.1236	0.3398	0.0354	54.6382	66.8958	0.4412	0.1430	0.6172	0.3617	19.1474
ISIMUD (10%)	0.0229	0.0358	0.1813	0.0038	34.1034	75.2347	0.3407	0.1395	0.3005	0.5840	21.0712
ISIMUD (30%)	**0.0150**	0.0540	0.1111	0.0038	23.0388	62.2054	0.4045	0.2490	0.4218	0.7242	23.6680
ISIMUD (50%)	0.0155	**0.0178**	0.1161	0.0031	22.6315	60.9141	0.4353	0.3480	0.4455	0.8020	24.4020
ISIMUD (70%)	0.0419	0.0836	**0.0066**	0.0033	22.1886	59.0470	0.5270	0.3312	0.5637	0.6957	26.0235
ISIMUD (100%)	0.0167	0.0384	0.0258	**0.0031**	**10.4256**	45.4230	0.5275	**0.5292**	0.6745	**0.8845**	**29.5449**

Table 10: Evaluation results for Weather Conceptual dataset.

Model	ACD	SD	KD	MDD	FID	J-FTSD	Precision	Recall	Joint Precision	Joint Recall	CTTP Score
Bridge	0.0343	19.5672	2320.6266	0.0205	45.0646	47.7810	0.7533	0.0292	0.6301	0.2497	27.3101
DiffuSETS	0.1379	19.1210	2317.8549	0.0395	149.2328	156.1811	0.5943	0.0025	0.1057	0.0368	16.0544
T2S	0.0932	18.8126	2315.7835	0.0303	110.0314	114.0980	0.1852	0.2594	0.2071	0.2154	20.2179
TEdit	0.0341	23.4401	2362.1122	0.0087	12.0106	14.0126	0.8595	0.4530	0.8321	0.7543	30.2036
Text2Motion	0.0500	18.1200	2298.9082	0.0077	5.5376	6.8965	**0.9215**	0.5861	0.9726	0.9174	**32.8174**
TimeVQVAE	0.0707	19.0337	2319.5249	0.0263	47.5908	53.5766	0.3864	0.0030	0.2642	0.1413	22.9355
TimeWeaver	0.0417	30.7571	2618.7799	0.0092	14.5267	16.7852	0.8554	0.3404	0.7838	0.6519	29.3724
TTSCGAN	0.1674	19.5566	2319.8547	0.0278	77.7704	87.3545	0.0988	0.0023	0.1047	0.0302	19.4969
VerbalTS	**0.0312**	**16.8770**	**2119.4812**	0.0069	6.7655	8.2458	0.9062	0.5810	0.9365	0.8643	32.0586
WaveStitch	0.0360	19.3397	2381.9672	0.0100	23.5472	25.8877	0.8214	0.3806	0.6773	0.5343	28.1586
ISIMUD (10%)	0.0984	19.0402	2316.2249	0.0079	15.0749	18.1609	0.5457	0.7713	0.7858	0.9390	29.3134
ISIMUD (30%)	0.0661	18.5097	2310.4206	0.0091	6.7772	8.2687	0.7912	0.8857	0.9825	0.9962	31.6196
ISIMUD (50%)	0.0665	18.6089	2308.3660	**0.0056**	3.6166	5.0473	0.8422	0.9718	0.9848	0.9977	31.8522
ISIMUD (70%)	0.0644	18.1543	2299.5348	0.0056	**2.2375**	**3.6268**	0.9123	**0.9748**	0.9909	0.9985	32.0879
ISIMUD (100%)	0.0407	17.2208	2210.4551	0.0070	3.3658	4.7553	0.9192	0.9733	**0.9916**	**0.9992**	32.1844

G.2 Question Answering

We report the category-level accuracy of question answering on the ChatTime (Wang et al., 2025) QA benchmark. Higher accuracy indicates better time-series understanding.

Table 11: Category-level accuracy for question answering on ChatTime. Higher is better. Best results are in bold and second-best results are underlined.

Model	Trend	Volatility	Seasonality	Outlier
GPT4	0.7015	0.4979	0.3515	0.6327
GPT3.5	0.5846	0.3839	0.3952	0.5940
GLM4	0.7574	0.4770	0.3464	0.5298
LLaMA3	0.5855	0.4756	0.3958	0.5694
ChatTime	<u>0.9068</u>	<u>0.6954</u>	<u>0.6517</u>	<u>0.9032</u>
IsimuD (Ours)	**0.9864**	**0.8852**	**0.6545**	**0.9635**

H Additional Analysis

H.1 Scaling of Pretraining Corpus Size

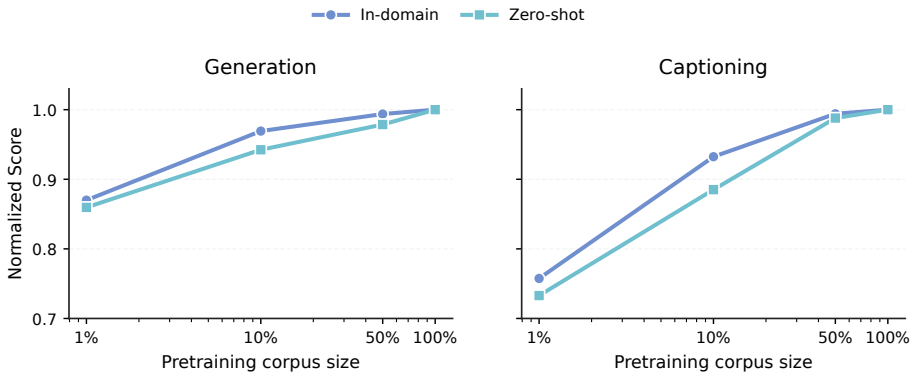

Figure 8: Scaling of pretraining corpus size on generation and captioning performance. We pretrain IsimuD using 1%, 10%, 50%, and 100% of Book-of-IsimuD. The left panel reports generation performance and the right panel reports captioning performance. Scores are normalized with respect to the corresponding 100% model.

We further study the effect of pretraining corpus size on the performance of IsimuD. Specifically, we pretrain four variants of IsimuD using 1%, 10%, 50%, and 100% of Book-of-IsimuD, respectively. For all variants, we keep the model architecture, optimization hyperparameters, and number of training epochs unchanged. The zero-shot datasets, weatherbench and fox are only used for evaluation.

We evaluate scaling behavior on both in-domain and zero-shot datasets. The in-domain group includes era5, ercot, city_bikes, covid_deaths, and favorita_transactions. The zero-shot group includes weatherbench and fox. For generation, we use J-FTSD and CTTP; for captioning, we use ROUGE-L and CTTP. Since these metrics have different scales and directions, we normalize each metric with respect to the corresponding 100% model before averaging across datasets.

For a dataset d and pretraining ratio r, the normalized generation score is defined as

$$s_d^{\text{gen}}(r) = \frac{1}{2}\left(\frac{\text{J-FTSD}_d(100\%)}{\text{J-FTSD}_d(r)} + \frac{\text{CTTP}_d^{\text{gen}}(r)}{\text{CTTP}_d^{\text{gen}}(100\%)}\right), \tag{25}$$

where lower J-FTSD and higher CTTP indicate better generation quality. Similarly, the normalized captioning score is defined as

$$s_d^{\mathrm{cap}}(r) = \frac{1}{2}\left(\frac{\text{ROUGE-L}_d(r)}{\text{ROUGE-L}_d(100\%)} + \frac{\mathrm{CTTP}_d^{\mathrm{cap}}(r)}{\mathrm{CTTP}_d^{\mathrm{cap}}(100\%)}\right). \tag{26}$$

We then average the normalized score over each dataset group:

$$S_{\mathcal{G}}(r) = \frac{1}{|\mathcal{G}|}\sum_{d\in\mathcal{G}} s_d(r), \qquad \mathcal{G} \in \{\mathcal{D}_{\mathrm{ID}}, \mathcal{D}_{\mathrm{ZS}}\}. \tag{27}$$

Fig. 8 presents the results. As the pretraining corpus size increases, both generation and captioning obtain higher normalized scores on in-domain datasets. The zero-shot curves also improve with larger pretraining corpora, showing that increasing the amount of paired data benefits generalization to unseen datasets. These results provide additional evidence that ISIMUD benefits from larger-scale pretraining and learns transferable representations across different time-series domains.

H.2 Significance Test for the Ablation Study

We extend the ablation study in Sec. 4.4 to all evaluated datasets and seeds. *w/o NBP* removes next-block prediction for captioning, while *w/o NSP* removes next-scale prediction for generation and directly generates the original-resolution series. *w/o Caption* and *w/o Generation* remove the corresponding training objectives, resulting in generation-only and captioning-only variants, respectively.

Table 12: Ablation results aggregated over all evaluated datasets and inference seeds. $\mathrm{CTTP}_{\mathrm{gen}}$ and $\mathrm{CTTP}_{\mathrm{cap}}$ denote CTTP scores for generation and captioning, respectively. † indicates that ISIMUD significantly outperforms every applicable ablation for the corresponding metric under paired permutation tests across datasets ($p < 0.05$). Best results are in bold.

	Generation		Captioning	
Variant	J-FTSD ↓	$\mathrm{CTTP}_{\mathrm{gen}}$ ↑	ROUGE-L ↑	$\mathrm{CTTP}_{\mathrm{cap}}$ ↑
ISIMUD	**1.369**†	**0.571**†	**0.441**†	**0.571**†
w/o NBP	1.472	0.555	0.353	0.267
w/o NSP	1.468	0.566	0.439	0.560
w/o Caption	1.471	0.566	—	—
w/o Generation	—	—	0.434	0.555

As shown in Tab. 12, ISIMUD achieves the best performance across all four metrics. Removing either task degrades the performance of the remaining task, supporting the benefit of unified training. Removing NBP substantially reduces captioning performance, while removing NSP degrades generation performance, further validating the modality-specific semi-autoregressive designs.

We further conduct paired permutation tests across datasets to evaluate whether these improvements are statistically significant. For each metric, † indicates that ISIMUD significantly outperforms every applicable ablation at $p < 0.05$. The significance criterion is satisfied for all four metrics.

I Case Studies

I.1 Generation

Fig. 9 presents qualitative examples of text-conditioned time series generation under the next-scale prediction paradigm. Each row provides a textual condition and compares the ground-truth series with samples generated at three scales: coarse, medium, and fine, as discussed in Appendix F.1. The results illustrate the progressive nature of ISIMUD: the coarse-scale generation first captures the low-frequency trajectory and broad regime changes, the medium-scale generation further recovers periodic structure and intermediate fluctuations, and the fine-scale generation adds local variations, sharp transitions, and short-term irregularities. Through this coarse-to-fine process, ISIMUD gradually refines the generated sequence so that it better matches the trend, periodicity, and local morphology specified by the text.

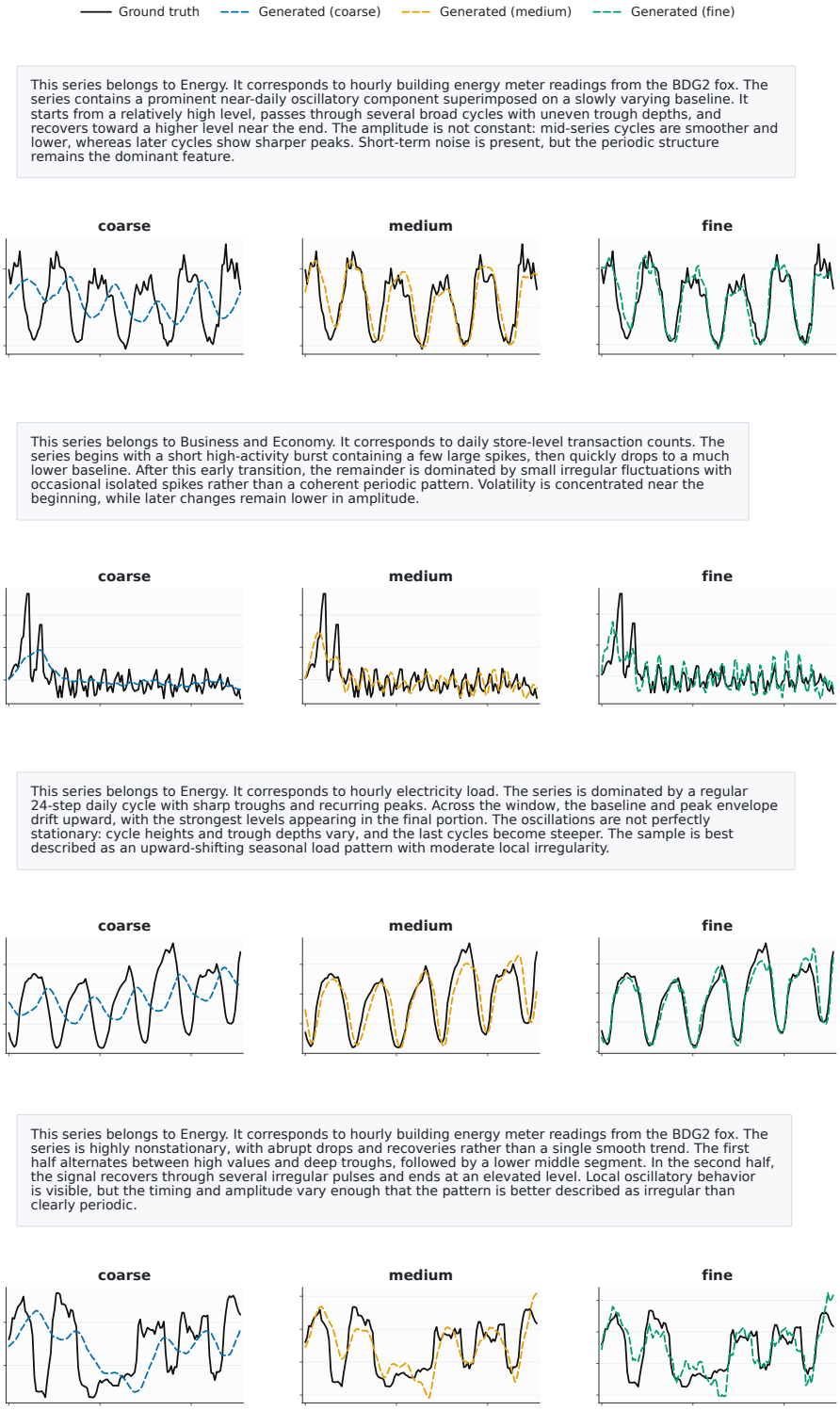

Figure 9: Case studies of text-conditioned generation. Each example shows the generated time series at three scales: coarse, medium, and fine.

I.2 CAPTIONING

Fig. 10 presents qualitative examples of time series captioning. In addition to describing the temporal morphology, ISIMUD correctly identifies the domain and dataset-level semantics of each input sequence. For example, the generated captions assign the samples to the appropriate domains, such as Weather and Health, and further specify the corresponding channel context, such as monthly tourism demand and daily store-level transaction counts. The generated captions also remain consistent with the observed morphology, including upward seasonal demand and irregular transaction dynamics. These examples show that ISIMUD can jointly produce domain-aware semantic descriptions and faithful morphology summaries.

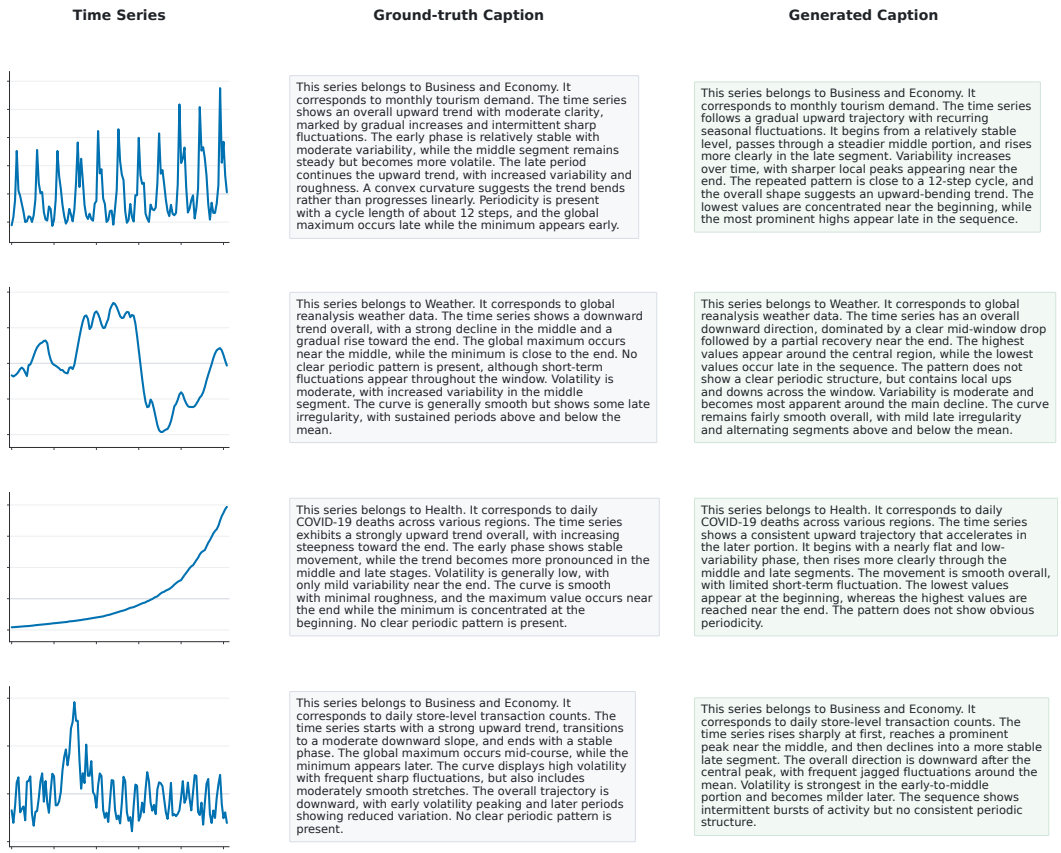

Figure 10: Case studies of time series captioning. Each row shows an input time series, the ground-truth caption, and the caption generated by ISIMUD.