Neo-InstructTime: Synthesizing Deceleration Events in Neonatal Vital Signs Using Natural Language

Anonymous Author(s)

Abstract

Decelerations in preterm infant vital signs (heart rate and oxygen saturation) are critical biomarkers of adverse outcomes in the NICU. Synthesizing such events can address data scarcity and support counterfactual reasoning on their pathophysiological signatures. Although recent advances in time series editing allow fine-grained modification of time series toward target conditions while preserving others, existing methods rely on rigid feature vectors and lack control over editing strength. We propose InstructTime, the first instruction-based time series editor, which specifies edits in natural language and enables controllable editing strength. Conditioning on free-form text enables incorporating nuanced clinical details such as demographics, comorbidities, interventions, and individualized information from medical free-text data. We evaluate InstructTime on editing decelerations in neonatal vital signs through four research questions: (1) performance comparison to state-of-the-art editors; (2) robustness under rare-event prevalence; (3) the effect of individualized clinical context on editing quality; and (4) exploratory insights from counterfactual edits. Our results show that InstructTime can synthesize realistic decelerations, maintain robust quality under data scarcity, and provide exploratory insights into these clinically significant deceleration events in neonatal vital signs.

1 Introduction

Bradycardia and oxygen desaturation are among the most clinically significant deceleration events in neonatal intensive care (NICU). Bradycardia is typically defined as heart rate below 100 bpm (or 80 bpm for severe cases) in preterm infants [16, 31, 8], and oxygen desaturation is usually defined as peripheral oxygen saturation (SpO_2) below 80–85% [9, 17]. Both events are critical biomarkers: their frequency, duration, and severity are strongly associated with poor neurodevelopmental outcomes and higher risk of morbidity and mobility in preterm infants [1, 12, 14, 21, 8, 18]. Continuously acquiring such deceleration events from an individual's bedside monitor requires dedicated time and labor, yet understanding their pathophysiological signatures remains of high interest and importance.

Recent work on time series generation has shown promise for synthesizing realistic data to support classification, forecasting, and counterfactual reasoning tasks [35, 28, 23, 30, 5, 25, 7, 2, 36, 13, 33, 27, 6, 3, 15]. Time series editing (TSE) is a fine-grained extension, which modifies an existing series to satisfy new conditions while preserving its original characteristics [11, 19, 37]. For instance (Figure 1), given a normal infant heart rate, a clinician may ask: "What if the infant experienced bradycardia, with heart rate falling below 80 or 100 bpm?" An editor would then generate a plausible trajectory specifying where, how severely, and in what shape such events occur, while preserving other conditions. The ability to editing a time series towards specified conditions is especially important for addressing data scarcity and enabling pathological counterfactual reasoning in medical research.

State-of-the-art diffusion-based TSE methods remain limited: they rely on handcrafted feature vectors as rigid condition format and overlook nuanced, individualized context in medical free-form data. They also yield all-or-nothing edits via sampling, whereas clinical settings may explore progressive edits—e.g., introducing deceleration events from mild to severe—for hypothesis generation.

We introduce **Instruction-based Time Series Editing**, a new task where an editor takes a time series and a natural language instruction specifying target conditions, and generates a modified time series reflecting those conditions. Unlike prior approaches with predefined attributes, this setting conditions

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

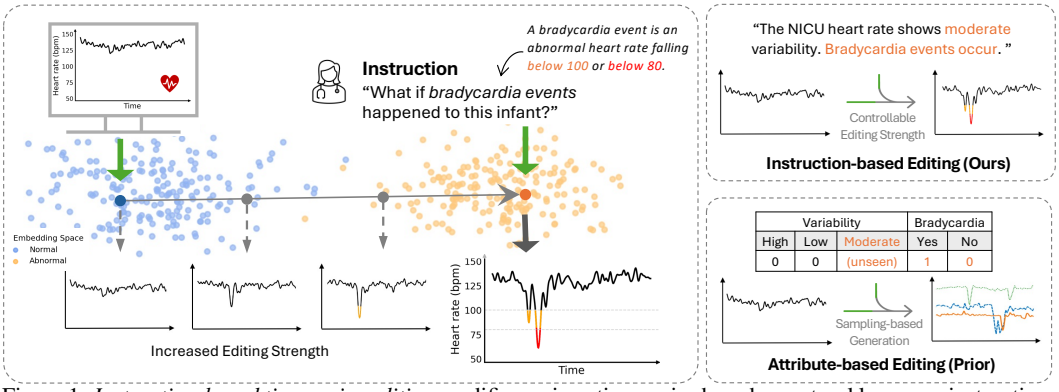

Figure 1: *Instruction-based time series editing* modifies a given time series based on natural language instructions with controllable editing strength. For example, medical researchers can use natural language to add abnormal deceleration events (bradycardia) to a normal heart rate with increasing severity.

on free-form text, capturing nuanced context—such as patient demographics, comorbidities, and interventions—when paired with clinical time series. To address challenges such as mapping multi-condition instructions to multi-resolution time-series patterns and enabling fine-grained control over edit strength, we propose InstructTime, the first instruction-based editor, which aligns time-series and text embeddings via contrastive learning and decodes interpolated embeddings for controllable edits.

In this paper, we focus on evaluating InstructTime on editing deceleration events in neonatal vital sign time series through the following **research questions**: (1) How does InstructTime perform compared to state-of-the-art methods? (2) How does real-world data scarcity, where deceleration events are rare, affect its editing quality? (3) Does adding individualized context in instructions enhance editing? (4) Can counterfactual edits offer deeper exploratory insights into decelerations in neonatal vital signs?

2 InstructTime: Instruction-based Time Series Editor

Let $\mathbf{x} \in \mathbb{R}^T$ be a time series with T timesteps, and let $\mathbf{c} = [c_1, \ldots, c_L]$ be a natural language instruction of L tokens describing the target condition. The task of instruction-based time series editing is to learn a function f_θ that generates an edited time series $\hat{\mathbf{x}} = f_\theta(\mathbf{x}, \mathbf{c})$ such that $\hat{\mathbf{x}}$ reflects the condition expressed in $\mathbf{c}$. We propose **InstructTime**, the first instruction-based time series editor, which leverages contrastive learning and an interpolated editing procedure for controllable strength. As shown in Figure 2, InstructTime consists of a multi-resolution time series encoder, an instruction encoder generalizable to diverse semantic expressions, and a conditional decoder that generates edited series by modeling intra- and inter-modality relationships with self-attention. Details of the model architecture and implementation are provided in the Appendix A.

Training. InstructTime is trained on paired data $\{(\mathbf{x}_i, \mathbf{c}_i)\}_{i=1}^N$ with a joint loss $\mathcal{L} = \mathcal{L}_{\text{contrast}} + \alpha \cdot \mathcal{L}_{\text{recon}}$, where $\mathcal{L}_{\text{contrast}}$ is the symmetric InfoNCE loss [22] encouraging alignment between $\mathbf{z}_x$ and $\mathbf{z}_c$ in the unit-length hypersphere, and $\mathcal{L}_{\text{recon}}$ is the mean squared error between $\mathbf{x}$ and $\hat{\mathbf{x}}$. Training proceeds in two stages: first optimizing $\mathcal{L}_{\text{contrast}}$ to align encoders, then minimizing $\mathcal{L}$ to train both encoders and the decoder. The weight α balances reconstruction against alignment.

Figure 2: Model architecture of InstructTime. In training, time series–description pairs are mapped to a shared hypersphere, then the decoder reconstructs the input series. In editing, interpolated embeddings of time series and instruction are decoded to generate edits of varying strength.

Interpolated Editing Procedure. To control editing strength, we interpolate between embeddings: $\mathbf{z}_w = (1-w)\mathbf{z}_x + w\mathbf{z}_c, \quad \hat{\mathbf{x}}_w = \Psi(\mathbf{z}_w, \mathbf{z}_c), \quad w \in [0,1]$. When $w = 0$, $\hat{\mathbf{x}}_w$ reconstructs the input, while $w = 1$ generates solely from the instruction. Increasing w blends stronger target conditions to the input time series, e.g., progressively intensifying bradycardia or SpO_2 desaturation events.

3 Synthesizing Deceleration Events in NICU Vital Signs

Datasets. We use a published dataset of daily vital sign observations from 2,964 infants admitted to the University of Virginia NICU between 2012–2016 [10, 29], consisting of 10-minute HR and SpO_2 segments (length 300, sampled every 2s). The processed **HR** dataset contains 36,679 series, including 2,147 bradycardia events (prevalence 0.06), defined as HR <100 bpm up to 300s [1]. The processed **SpO_2** dataset contains 30,000 series, balanced to 15,000 desaturation events and 15,000 controls (due to high prevalence), where desaturation is defined as SpO_2 <90% for 10–300s [34]. For both vital signs, a valid event requires a negative drop rate prior to onset and a positive recovery afterward. Each series is labeled as either "No events." or "Desaturation/Bradycardia events happened." All series are z-normalized and partitioned into training, validation, and held-out test sets (70–20–10).

Compared methods. We compare InstructTime with two state-of-the-art editors: TimeWeaver [19] and TEdit [11], both implemented with the diffusion-based editing procedure from [11]. Since our formulation does not require original attributes as input, we adopt the denoising diffusion probabilistic model sampler from [11], which only conditions on target attributes.

Evaluation metrics. We evaluate editing quality of *adding or removing deceleration events* in HR and SpO_2 using three metrics: (1) Dynamic time warping [26] distance decrease, calculated as Δ**DTW**$= \text{median}_{\mathbf{x}_{\tilde{\mathbf{c}}} \in \mathcal{D}_{\tilde{\mathbf{c}}}}[\text{DTW}(\hat{\mathbf{x}}, \mathbf{x}_{\tilde{\mathbf{c}}}) - \text{DTW}(\mathbf{x}, \mathbf{x}_{\tilde{\mathbf{c}}})]$, measures how much closer the edited series $\hat{\mathbf{x}}$ is to real observations $\mathcal{D}_{\tilde{\mathbf{c}}}$ with target condition than the input $\mathbf{x}$, where lower (more negative) values indicate better editing. (2) Longest common subsequence [4] similarity increase, calculated as Δ**LCSS**$= \text{median}_{\mathbf{x}_{\tilde{\mathbf{c}}} \in \mathcal{D}_{\tilde{\mathbf{c}}}}[\text{LCSS}(\hat{\mathbf{x}}, \mathbf{x}_{\tilde{\mathbf{c}}}) - \text{LCSS}(\mathbf{x}, \mathbf{x}_{\tilde{\mathbf{c}}})]$, measures how much more similar $\hat{\mathbf{x}}$ is to $\mathcal{D}_{\tilde{\mathbf{c}}}$ than $\mathbf{x}$, where higher values indicate better editing. (3) Log Ratio of Target-to-Source [11], calculated as **RaTS**$= \log\left(\frac{p(\tilde{\mathbf{c}}|\hat{\mathbf{x}})}{p(\tilde{\mathbf{c}}|\mathbf{x})}\right)$, where $p(\tilde{\mathbf{c}} \mid \mathbf{x})$ is estimated by a multi-class classifier trained on a held-out dataset, measures the relative increase in likelihood of the target attribute $\tilde{\mathbf{c}}$ in the edits.

RQ1: Editing quality benchmark. We benchmark editors in both instruction-based and attribute-based settings. For instruction-based editing, we adapt prior methods by replacing categorical vectors with text instruction embeddings in their condition encoders. In the attribute-based setting, we input categorical attribute vectors directly into InstructTime's instruction encoder to match prior methods. Table 1 shows that InstructTime is a state-of-the-art editor for synthesizing deceleration events in NICU vital signs under both settings. Examples of deceleration edits are shown in Figure 3.

	HR Bradycardia			SpO_2 Desaturation		
	ΔDTW ↓	ΔLCSS ↑	RaTS ↑	ΔDTW ↓	ΔLCSS ↑	RaTS ↑
Time Weaver	-2.66	0.02	0.01	-4.02	0.11	0.15
TEdit	-5.34	0.13	0.02	-5.11	0.12	0.15
InstructTime	**-10.54**	**0.29**	**0.18**	**-8.20**	**0.16**	**0.34**

(a) Instruction-based editing

	HR Bradycardia			SpO_2 Desaturation		
	ΔDTW ↓	ΔLCSS ↑	RaTS ↑	ΔDTW ↓	ΔLCSS ↑	RaTS ↑
Time Weaver	-9.23	0.19	0.20	-6.32	0.14	0.28
TEdit	**-12.91**	0.21	0.52	-5.83	0.13	0.28
InstructTime	-11.18	**0.24**	**0.72**	**-8.63**	**0.16**	**0.69**

(b) Attribute-based editing

Table 1: Benchmark comparison of time series editors for editing decelerations in neonatal vital signs.

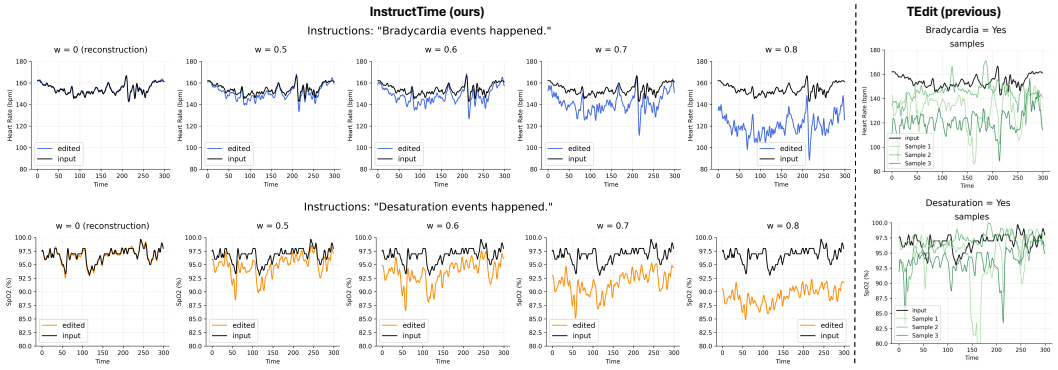

Figure 3: Examples of adding decelerations (bradycardia or desaturation) to normal HR or SpO_2 from NICU. InstructTime uses natural language instructions and controls deceleration severity via editing strength w, while prior diffusion-based method TEdit rely on categorical attributes and generate diverse edits via sampling.

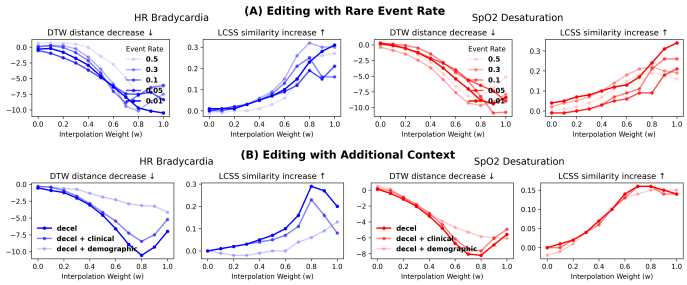

Figure 4: Impact on InstructTime's editing quality (ΔDTW$\downarrow$ and ΔLCSS$\uparrow$) by two conditions: (A) rare prevalence of deceleration at varying rates; and (B) adding nuanced patient information to deceleration description (decel) in the instruction, including clinical (death outcome, usage of supplemental oxygen) and demographic (gestational age, Apgar score, gender, race, ethnicity) information.

RQ2: Editing under data scarcity. To evaluate the real-world utility of InstructTime in medical research, we first assess the impact of rare-event prevalence on editing quality by downsampling the no-event HR series (or the event-group SpO_2 series, due to high prevalence) to achieve deceleration rates of 0.5, 0.3, 0.1, 0.05, and 0.01. As shown in Figure 4(A), InstructTime maintained robust performance in adding or removing decelerations in neonatal vital signs.

RQ3: Editing with nuanced clinical context. We also assess whether adding nuanced clinical context to the instruction, alongside the deceleration event description, improves InstructTime's editing quality. As shown in Figure 4(B), adding clinical intervention (usage of supplemental oxygen) and clinical outcome (death in 7 days) yields similar editing performance to only describing the deceleration in the instruction. However, adding patient demographic information (gestational age, gender, race, ethnicity, and Apgar scores [20]) generally degrades editing quality. We speculate this may be because overly personalized instructions can reside out-of-distribution in the shared embedding space (e.g., decelerations may rarely happen to infants with larger gestational age or from certain demographic population). This result also suggests that adjusting current models to account for multi-level structure—where time series from the same patient share similar temporal patterns while those across patients are independent—warrants further investigation.

RQ4: Inspecting diverse deceleration phases in counterfactual edits. Normal neonatal HR is 120–160 bpm with 5–25 bpm oscillations, and SpO_2 is 90–100% with 2–3% fluctuations [1]. To better understand how deceleration events would appear in normal HR or SpO_2, we adopt a paired shapelet classification procedure (details in Appendix B) to detect discriminative 0.5-1 minute subsequences (shapelets) that represent decelerations in counterfactual series edited by InstructTime versus their inputs. For comparison, we applied the same shapelet detection to randomly sampled real deceleration time series versus normal ones. As shown in Figure 5, the top shapelets detected in observed HR converged to a unique pattern: a 1-minute sudden drop from around 150 bpm to below 80, followed by recovery. In contrast, in the edited series where InstructTime introduced bradycardia into normal HR, the discriminative subsequence varies with the baseline. With a high baseline (above 150 bpm), the editor produced a similar severe drop, while with a lower baseline (e.g., <120 bpm), it generated a global downward shift with shallower, multi-phasic events. This diversity of shapelets aligns with clinical empirical observations [8]. Similarly, in SpO_2, the editor generally produced desaturation with a global baseline downshift around 90%. Counterfactual edits also suggested additional shapelets, including sharp drops from 90% to 80% and longer turbulent episodes below 90%, compared to the single uni-phase drop detected in empirical observations.

4 Conclusion

In summary, we propose InstructTime, the first instruction-based time series editor, enabling natural language–driven synthesis of clinically significant decelerations in neonatal vital signs with controllable strength. It shows robust performance under rare-event scarcity, incorporates clinical context, and produces counterfactual edits that offer pathophysiological insights, with future work on multi-level modeling, uncertainty estimation, and broader medical applications.

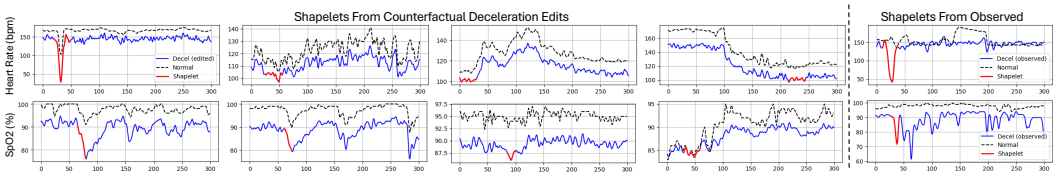

Figure 5: For HR and SpO_2, multi-phase discriminative subsequences (shapelets) of deceleration patterns appear in counterfactual edits (left), whereas a uni-phase deceleration pattern is observed empirically (right).

References

[1] Namasivayam Ambalavanan, Debra E Weese-Mayer, Anna Maria Hibbs, Nelson Claure, John L Carroll, J Randall Moorman, Eduardo Bancalari, Aaron Hamvas, Richard J Martin, Juliann M Di Fiore, et al. Cardiorespiratory monitoring data to predict respiratory outcomes in extremely preterm infants. *American journal of respiratory and critical care medicine*, 208(1):79–97, 2023.

[2] Gaby Baasch, Guillaume Rousseau, and Ralph Evins. A conditional generative adversarial network for energy use in multiple buildings using scarce data. *Energy and AI*, 5:100087, 2021.

[3] Kasun Bandara, Hansika Hewamalage, Yuan-Hao Liu, Yanfei Kang, and Christoph Bergmeir. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognition*, 120:108148, 2021.

[4] Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, pages 39–48. IEEE, 2000.

[5] Defu Cao, Wen Ye, Yizhou Zhang, and Yan Liu. Timedit: General-purpose diffusion transformers for time series foundation model. September 2024.

[6] Muxi Chen, Zhijian Xu, Ailing Zeng, and Qiang Xu. Fraug: Frequency domain augmentation for time series forecasting. *arXiv preprint arXiv:2302.09292*, 2023.

[7] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. November 2021.

[8] Matthieu Doyen, Alfredo I Hernández, Cyril Flamant, Antoine Defontaine, Géraldine Favrais, Miguel Altuve, Bruno Laviolle, Alain Beuchée, Guy Carrault, and Patrick Pladys. Early bradycardia detection and therapeutic interventions in preterm infant monitoring. *Scientific Reports*, 11(1):10486, 2021.

[9] Karen D Fairchild, V Peter Nagraj, Brynne A Sullivan, J Randall Moorman, and Douglas E Lake. Oxygen desaturations in the early neonatal period predict development of bronchopulmonary dysplasia. *Pediatric research*, 85(7):987–993, 2019.

[10] Ian German Mesner. Pediatric Academic Societies 2024 NICU Mortality Prediction Challenge, 2024.

[11] Baoyu Jing, Shuqi Gu, Tianyu Chen, Zhiyu Yang, Dongsheng Li, Jingrui He, and Kan Ren. Towards editing time series. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[12] Sherry L Kausch, Jackson G Brandberg, Jiaxing Qiu, Aneesha Panda, Alexandra Binai, Joseph Isler, Rakesh Sahni, Zachary A Vesoulis, J Randall Moorman, Karen D Fairchild, et al. Cardiorespiratory signature of neonatal sepsis: development and validation of prediction models in 3 nicus. *Pediatric research*, 93(7):1913–1921, 2023.

[13] Delanyo Kwame Bensah Kulevome, Hong Wang, Bernard Mawuli Cobbinah, Ernest Smith Mawuli, and Rajesh Kumar. Effective time-series data augmentation with analytic wavelets for bearing fault diagnosis. *Expert Systems with Applications*, 249:123536, 2024.

[14] Lisa Letzkus, Robin Picavia, Genevieve Lyons, Jackson Brandberg, Jiaxing Qiu, Sherry Kausch, Doug Lake, and Karen Fairchild. Heart rate patterns predicting cerebral palsy in preterm infants. *Pediatric Research*, pages 1–7, 2023.

[15] Michelle M Li, Kevin Li, Yasha Ektefaie, Shvat Messica, and Marinka Zitnik. Controllable sequence editing for counterfactual generation. *arXiv preprint arXiv:2502.03569*, 2025.

[16] Scott A Lorch, Lakshmi Srinivasan, and Gabriel J Escobar. Epidemiology of apnea and bradycardia resolution in premature infants. *Pediatrics*, 128(2):e366–e373, 2011.

[17] C McClure, S Young Jang, and Karen Fairchild. Alarms, oxygen saturations, and spo2 averaging time in the nicu. *Journal of neonatal-perinatal medicine*, 9(4):357–362, 2016.

[18] V Peter Nagraj, Robert A Sinkin, Douglas E Lake, J Randall Moorman, and Karen D Fairchild. Recovery from bradycardia and desaturation events at 32 weeks corrected age and nicu length of stay: an indicator of physiologic resilience? *Pediatric research*, 86(5):622–627, 2019.

[19] Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep Chinchali. Time weaver: A conditional time series generation model. *arXiv preprint arXiv:2403.02682*, 2024.

[20] American Academy of Pediatrics Committee on Fetus, Newborn, American College of Obstetricians, Gynecologists Committee on Obstetric Practice, Kristi L Watterberg, Susan Aucott, William E Benitz, James J Cummings, Eric C Eichenwald, Jay Goldsmith, Brenda B Poindexter, Karen Puopolo, et al. The apgar score. *Pediatrics*, 136(4):819–822, 2015.

[21] Jiaxing Qiu, Juliann M Di Fiore, Narayanan Krishnamurthi, Premananda Indic, John L Carroll, Nelson Claure, James S Kemp, Phyllis A Dennery, Namasivayam Ambalavanan, Debra E Weese-Mayer, et al. Highly comparative time series analysis of oxygen saturation and heart rate to predict respiratory outcomes in extremely preterm infants. *Physiological measurement*, 45(5):055025, 2024.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[23] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8857–8868. PMLR, 18–24 Jul 2021.

[24] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[25] Lei Ren, Haiteng Wang, and Yuanjun Laili. Diff-mts: Temporal-augmented conditional diffusion-based aigc for industrial time series toward the large model era. *IEEE Transactions on Cybernetics*, 54(12):7187–7197, 2024.

[26] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 2003.

[27] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. Data augmentation for univariate time series forecasting with neural networks. *Pattern Recognition*, 134:109132, 2023.

[28] Alexander Sommers, Logan Cummins, Sudip Mittal, Shahram Rahimi, Maria Seale, Joseph Jaboure, and Thomas Arnold. A survey of transformer enabled time series synthesis. June 2024.

[29] Brynne A Sullivan, Alvaro G Moreira, Ryan M McAdams, Lindsey A Knake, Ameena Husain, Jiaxing Qiu, Avinash Mudireddy, Abrar Majeedi, Wissam Shalish, Douglas E Lake, et al. Comparing machine learning techniques for neonatal mortality prediction: insights from a modeling competition. *Pediatric research*, pages 1–7, 2024.

[30] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24804–24816. Curran Associates, Inc., 2021.

[31] Henriëtte A van Zanten, Ratna NGB Tan, Martha Thio, JM De Man-Van Ginkel, EW Van Zwet, Enrico Lopriore, and Arjan B te Pas. The risk for hyperoxaemia after apnoea, bradycardia and hypoxaemia in preterm infants. *Archives of Disease in Childhood-Fetal and Neonatal Edition*, 99(4):F269–F273, 2014.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[33] Claudio Meneses Villegas, Jorge Littin Curinao, David Coo Aqueveque, Juan Guerrero-Henríquez, and Martín Vargas Matamala. Data augmentation and hierarchical classification to support the diagnosis of neuropathies based on time series analysis. *Biomedical Signal Processing and Control*, 95:106302, 2024.

[34] Debra E Weese-Mayer, Juliann M Di Fiore, Douglas E Lake, Anna Maria Hibbs, Nelson Claure, Jiaxing Qiu, Namasivayam Ambalavanan, Eduardo Bancalari, James S Kemp, Amanda M Zimmet, et al. Maturation of cardioventilatory physiological trajectories in extremely preterm infants. *Pediatric research*, 95(4):1060–1069, 2024.

[35] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[36] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[37] Hao Yu, Chu Xin Cheng, Runlong Yu, Yuyang Ye, Shiwei Tong, Zhaofeng Liu, and Defu Lian. How to unlock time series editing? diffusion-driven approach with multi-grained control. *arXiv preprint arXiv:2506.05276*, 2025.

A InstructTime Architecture and Implementation

Multi-resolution encoder $\mathcal{E}_\phi$. The encoder $\mathcal{E}_\phi$ maps $\mathbf{x}$ into an embedding $\mathbf{z}_x$ normalized on the unit hypersphere. To capture both global trends and local fluctuations, $\mathcal{E}_\phi$ uses k parallel 1D CNNs with kernel sizes proportional to T, each producing a d-dimensional representation. These are concatenated into $\mathbf{z}_x \in \mathbb{R}^D$, where $D = k \cdot d$.

Instruction encoder $\mathcal{E}_\theta$. The encoder $\mathcal{E}_\theta$ maps $\mathbf{c}$ into $\mathbf{z}_c$, also normalized on the hypersphere. A pretrained text model (*paraphrase-mpnet-base-v2* [24]) first transforms instructions into numeric features, which are then processed by k parallel MLPs to align with the resolutions of $\mathcal{E}_\phi$. The concatenated embedding $\mathbf{z}_c$ thus aligns condition semantics with temporal resolutions of $\mathbf{z}_x$.

Conditional decoder Ψ. Given $(\mathbf{z}_x, \mathbf{z}_c)$, the decoder Ψ generates $\hat{\mathbf{x}} = \Psi(\mathbf{z}_x, \mathbf{z}_c)$. Built on Transformer self-attention [32], Ψ models dependencies within each modality and across modalities, mapping semantic conditions to appropriate time series patterns. The output state corresponding to the time series token is passed through a linear head to produce a new time series $\hat{\mathbf{x}} \in \mathbb{R}^T$.

Implementation details. For InstructTime, the time series encoder uses $k = 8$ parallel CNNs with kernel sizes proportional to the time series length T, using fractions $1, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{10}$. Larger fractions $(1, \frac{2}{3}, \frac{1}{2})$ are used to capture global properties such as trend and seasonality, while smaller fractions are intended to encode localized patterns such as abrupt shifts in mean and local variability. Text instructions are encoded using the pretrained SentenceTransformer model *paraphrase-mpnet-base-v2* [24], which provides effective embedding of rich semantic meaning from multiple sentences and is efficient to compute. The resulting vectors are then processed by $k = 8$ parallel MLPs to generate instruction embeddings $\mathbf{z}_\mathrm{c}$. The embeddings $\mathbf{z}_\mathrm{x}$ and $\mathbf{z}_\mathrm{c}$ have dimension $D = 768$. When training InstructTime, we jointly optimize a contrastive loss and a reconstruction loss, with their relative contributions to the total loss controlled by a balancing weight α. In our experiments, α is set so that the unit contribution of the reconstruction loss is $10^{-\gamma}$ relative to the contrastive loss, with γ set to 1 by default. InstructTime is trained on all datasets using a single NVIDIA V100 GPU. Training completed within 1 hour for each dataset.

B Interpreting Counterfactual Edits

B.1 Paired Shapelet Classification

To interpret how counterfactual edits differ from the original time series, we employ a **paired shapelet classification** approach. We consider aligned pairs of raw inputs $X_{\text{raw}} \in \mathbb{R}^{n \times T}$ and their edited counterparts $X_{\text{edit}} \in \mathbb{R}^{n \times T}$, where $(x^i_{\text{raw}}, x^i_{\text{edit}})$ denotes the raw–edited pair for sample $i \in \{1, \ldots, n\}$. Candidate shapelets $s \in \mathbb{R}^\ell$ are extracted from the edited series, and their discriminative value is measured by comparing distances to both members of each pair. For sample i, the paired distance difference is defined as

$$\Delta^i_d = d(s, x^i_{\text{raw}}) - d(s, x^i_{\text{edit}}),$$

where $d(s, x)$ denotes the minimum Euclidean distance between s and any subsequence of x of length ℓ. The discriminative strength of a shapelet is quantified by a one-direction signal-to-noise ratio (SNR):

$$SNR(s) = \max\left(0, \frac{\mu(\Delta_d)}{\sqrt{\sigma^2(\Delta_d) + \epsilon}}\right),$$

where $\mu(\Delta_d)$ and $\sigma^2(\Delta_d)$ denote the mean and variance of the paired distance differences across samples. This formulation ensures that only shapelets with $\mu(\Delta_d) > 0$—that is, subsequences consistently closer to the edited time series than to the raw time series—receive a positive score. Shapelets with $\mu(\Delta_d) \leq 0$ are discarded, as they do not represent patterns introduced by the editing process. This paired scoring directly highlights temporal discriminative subsequences that are introduced by the editor. Compared to traditional unpaired shapelet discovery, the paired formulation reduces variance and computing cost, removes the need for threshold tuning, and produces interpretable subsequences that reveal how counterfactual time series editing alters temporal structure.

Furthermore, the reliability of a shapelet in distinguishing between raw and counterfactual time series can be evaluated using the SNR. A high SNR indicates that the shapelet consistently matches one side more closely across pairs, with a large average gap and little variability, making it a strong and

reliable discriminator. A low SNR, by contrast, reflects either a small average difference or high variability across pairs, suggesting that the shapelet provides weak or unstable discrimination.

B.2 Implementation Details

We use a sliding-window approach to detect discriminative subsequences (shapelets) that separate raw from edited time series. Candidate shapelets are extracted from the augmented series, each defined by a length of 30–60 time steps (corresponding to 1–2 minutes of HR and SpO_2) and a stride of 5 steps. For each candidate, we compute the minimum Euclidean distance to all series in both raw and augmented sets. Candidates are scored using a signal-to-noise ratio (SNR) that favors patterns closer to the augmented series than to the raw series. The top candidates with the highest positive scores are retained as representative patterns of the edited series.