Rethinking Tokenization for Clinical Time Series: When Less is More

Rafi Al Attrach* RAFIAA@MIT.EDU
Massachusetts Institute of Technology (MIT), USA; Technical University of Munich (TUM), Germany

Rajna Fani* RAJNAF@MIT.EDU
Massachusetts Institute of Technology (MIT), USA; Technical University of Munich (TUM), Germany

David Restrepo DAVID.RESTREPO@CENTRALESUPELEC.FR
MICS, CentraleSupélec – Université Paris-Saclay, France

Yugang Jia YUGANG@MIT.EDU
Massachusetts Institute of Technology (MIT), USA

Leo Anthony Celi† LCELI@MIT.EDU
Massachusetts Institute of Technology (MIT), USA; Harvard Medical School, USA; Beth Israel Deaconess Medical Center, USA

Peter Schüffler† PETER.SCHUEFFLER@TUM.DE
Institute of Pathology, Technical University of Munich, Germany; Munich Center for Machine Learning (MCML), Germany

Abstract

Tokenization strategies shape how models process electronic health records, yet fair comparisons of their effectiveness remain limited. We present a systematic evaluation of tokenization approaches for clinical time series modeling using transformer-based architectures, revealing task-dependent and sometimes counterintuitive findings about temporal and value feature importance. Through controlled ablations across four clinical prediction tasks on MIMIC-IV, we demonstrate that explicit time encodings provide no consistent statistically significant benefit for the evaluated downstream tasks. Value features show task-dependent importance, affecting mortality prediction but not readmission, suggesting code sequences alone can carry sufficient predictive signal. We further show that frozen pretrained code encoders dramatically outperform their trainable counterparts while requiring dramatically fewer parameters. Larger clinical encoders provide consistent improvements across tasks, benefiting from frozen embeddings that eliminate computational overhead. Our controlled evaluation enables fairer tokenization comparisons and demonstrates that simpler, parameter-efficient approaches can, in many cases, achieve strong performance, though the optimal tokenization strategy remains task-dependent.

Keywords: tokenization, clinical time series, electronic health records, transformer models, healthcare AI

Data and Code Availability We use the publicly available MIMIC-IV dataset (Johnson et al., 2023), processed into the Medical Event Data Standard (MEDS) format (McDermott et al., 2025) using the MIMIC-IV MEDS ETL pipeline (Medical Event Data Standard Organization, 2024). Our experiments are conducted using the MEDS-Torch framework (Oufattole et al., 2024). Code for reproducing our experiments is available at https://github.com/rafiattrach/rethinking-ehr-tokenization.

Institutional Review Board (IRB) This research uses the publicly available MIMIC-IV dataset, which has been previously approved for research use and does not require additional IRB approval for this analysis.

* Shared first author.
† Shared corresponding author.

© R.A. Attrach, R. Fani, D. Restrepo, Y. Jia, L.A. Celi & P. Schüffler.

1. Introduction

Foundation models have transformed clinical machine learning (Li et al., 2020; Rasmy et al., 2021). Tokenization, the process of converting raw electronic health record (EHR) events into dense vector representations, is a fundamental step that affects how models process information (Ali et al., 2024). However, a key question persists: how should we tokenize complex, irregular medical time series data for optimal model performance? Unlike natural language or images, clinical data presents unique challenges: sparse measurements, irregular sampling, and heterogeneous feature types that demand specialized tokenization strategies. Recent benchmarks like EHRSHOT (Wornow et al., 2023) have highlighted the importance of systematic evaluation frameworks for clinical AI models, while generative approaches (Waxler et al., 2025; Pang et al., 2024) and foundation model applications (Renc et al., 2025) continue to expand the scope of clinical AI.

Recent frameworks like MEDS-Torch (Oufattole et al., 2024) have proposed multiple tokenization approaches, with Triplet (structured code-time-value triplets) and TextCode (natural language descriptions with structured features) representing two different philosophies. The Triplet approach, building on STraTS (Tipirneni and Reddy, 2022), represents each event as the sum of a learned code embedding, a time embedding, and a value embedding. In contrast, TextCode uses pretrained language models to encode code descriptions into semantic embeddings. While Triplet achieved the highest win rate and TextCode the lowest in initial benchmarks, these comparisons were potentially confounded by experimental design choices.

Despite recent innovations, establishing fair comparisons between tokenization strategies remains challenging. Prior benchmarks reported superior performance of Triplet-based representations over TextCode (Oufattole et al., 2024); however, these evaluations confounded several factors, including incomplete code-to-description mappings, the reliance on a single model size, and whether pretrained language encoders were frozen or fine-tuned. Such methodological inconsistencies make it difficult to attribute observed performance to the tokenization strategy itself rather than to experimental design choices.

To address these limitations, we conduct a series of systematic and controlled experiments designed to isolate the contribution of individual components. Our results challenge prevailing assumptions about clinical time series modeling, demonstrating that many widely adopted tokenization pipelines may be unnecessarily complex for certain tasks. By disentangling the effects of time encoding, value features, and embedding strategies, we enable clearer comparisons between tokenization approaches and highlight opportunities for simpler, more parameter-efficient models that maintain competitive performance.

2. Methods

2.1. Clinical Tasks and Dataset

Our evaluation uses four clinical prediction tasks from the MEDS-ACES task suite (Xu et al., 2024): in-hospital mortality within 24 hours of admission, ICU mortality within 24 hours of ICU stay, 1-year post-discharge mortality, and 30-day readmission. We use MIMIC-IV (Johnson et al., 2023), processed into the Medical Event Data Standard (MEDS) format (McDermott et al., 2025) using the MIMIC-IV MEDS ETL pipeline (Medical Event Data Standard Organization, 2024). This standardized format enables reproducible comparisons across different tokenization approaches.

2.2. Tokenization Approaches

We compare two tokenization strategies that represent different philosophies for encoding clinical events. **Triplet tokenization** represents each clinical event as the sum of three embeddings:

$$\mathbf{e}_{\text{triplet}} = \mathbf{W}_{\text{code}}\mathbf{c} + \text{CVE}_{\text{time}}(\mathbf{t}) + \text{CVE}_{\text{value}}(\mathbf{v}) \quad (1)$$

where $\mathbf{W}_{\text{code}}$ is a learned embedding matrix for medical codes $\mathbf{c}$, and CVE (Continuous Value Embedding) (Tipirneni and Reddy, 2022) maps scalar time deltas $\mathbf{t}$ and numeric values $\mathbf{v}$ to dense representations.

TextCode replaces the learned code embedding with representations from pretrained language models:

$$\mathbf{e}_{\text{textcode}} = \text{LM}(\text{desc}(\mathbf{c})) + \text{CVE}_{\text{time}}(\mathbf{t}) + \text{CVE}_{\text{value}}(\mathbf{v}) \quad (2)$$

where $\text{desc}(\cdot)$ maps medical codes to natural language descriptions and LM represents a pretrained language model encoder.

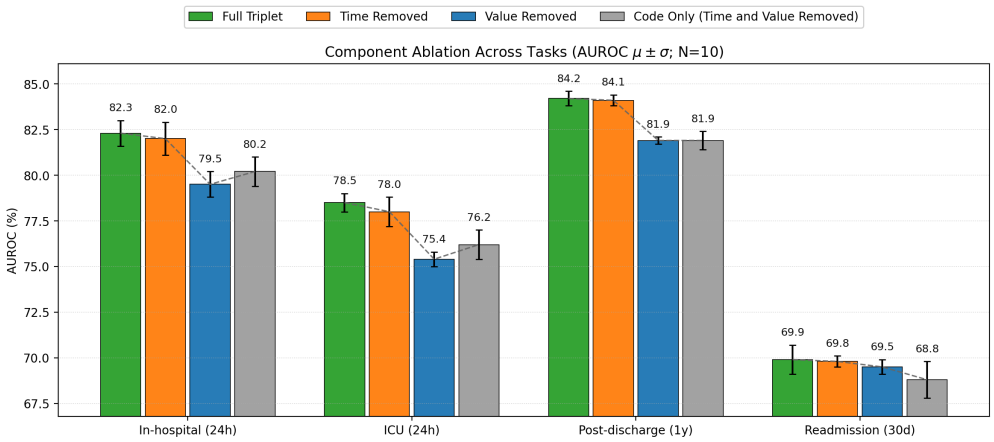

Figure 1: Triplet component ablation across four tasks. Error bars show ±sd over $N = 10$ seeds. Full numeric results are deferred to Appendix Table 2.

2.3. Controlled Experimental Design

To enable fair comparisons, we systematically vary TextCode along four controlled axes: 1) mapping coverage, 2) training approach, 3) encoder scale, and 4) encoder domain. For mapping coverage, we compare original sparse mappings (25% coverage) against enhanced mappings (100% coverage) that we filled using the MIMIC-IV MCP Server (Attrach et al., 2025). Here, coverage denotes the fraction of vocabulary codes with human-readable descriptions (rather than fallback identifiers); the original mapping leaves most codes without descriptions, whereas the enhanced mapping provides descriptions for all codes (see Appendix, Section A.1). This is critical because TextCode relies on natural language descriptions as input to language models (Su et al., 2025; Hegselmann et al., 2025). The training approach varies between trainable and frozen language model encoders. For encoder scale, we test models ranging from 15M to nearly 600M parameters. For encoder domain, we contrast clinical versus general-domain code encoders at comparable scale (both frozen) to isolate domain effects.

We conduct comprehensive component ablations on Triplet tokenization to understand the individual contribution of each modality. By systematically removing time features, value features, or both components while preserving the code information, we isolate the predictive signal contributed by each tokenization component across all clinical tasks.

2.4. Statistical Analysis

All models use supervised training in the standardized MEDS-Torch pipeline with transformer encoders. We report AUROC (Area Under the Receiver Operating Characteristic curve) as the primary metric (McDermott et al., 2024), aggregated over 10 random seeds. Statistical significance testing uses paired, two-sided Wilcoxon signed-rank tests, chosen for their robustness to non-normal distributions common in clinical prediction metrics. Bonferroni correction adjusts for multiple comparisons within each experimental group, with seed intersections used for proper pairing.

3. Results

3.1. Triplet component importance

A central finding is the limited impact of explicit time and value features for the evaluated tasks (as shown in Figure 1). Removing time features shows no statistically significant effect across all clinical tasks tested, suggesting that explicit temporal encoding may provide less benefit than commonly assumed for these prediction tasks with transformer architectures. Remarkably, for readmission prediction, even removing value features shows no significant impact, indicating that event code sequences alone carry sufficient predictive signal. While value features do contribute to mortality prediction tasks,

the consistent lack of temporal signal across all evaluated tasks reveals potential optimization opportunities in current tokenization approaches. Interestingly, for in-hospital and ICU mortality, the code-only model (with both time and value removed) outperformed the no-value variant, suggesting that isolated time features may introduce noise when value information is absent.

3.2. TextCode: mapping, freezing, and code encoders

Freezing the code encoder dramatically improves performance, and larger clinical code encoders improve further; this performance gain was statistically significant across all four tasks evaluated. BioClinical-ModernBERT-large substantially outperforms the baseline Tiny-ClinicalBERT model, while general-domain encoders like Qwen3-Embedding-0.6B achieve competitive but not superior performance compared to clinical models. Enhanced mapping coverage shows minimal impact compared to these architectural choices. Table 1 summarizes post-discharge mortality, which is representative of the trends observed across other tasks (see Appendix for full results).

Axis definitions. Each axis isolates a single experimental change against a controlled baseline configuration to isolate its effect:

- **Mapping**: Original vs. Enhanced description coverage (using trainable Tiny-ClinicalBERT).
- **Trainability**: Trainable vs. Frozen Tiny-ClinicalBERT code encoder (using Enhanced mapping).
- **Size**: Tiny-ClinicalBERT (≈15M) vs. BioClinical-ModernBERT-L (≈396M) (both frozen).
- **Domain**: Clinical (BioClinical-ModernBERT-L) vs. General (Qwen3, ≈596M) (both frozen).

3.3. Learnable time encodings

Given that simple time encoding seemed ineffective, we also tested more complex learnable time encoders including Time2Vec (Kazemi et al., 2019) and LeTE (Chen et al., 2025), but **they do not surpass the simple linear time feature**. Details are in Appendix Section A.4.

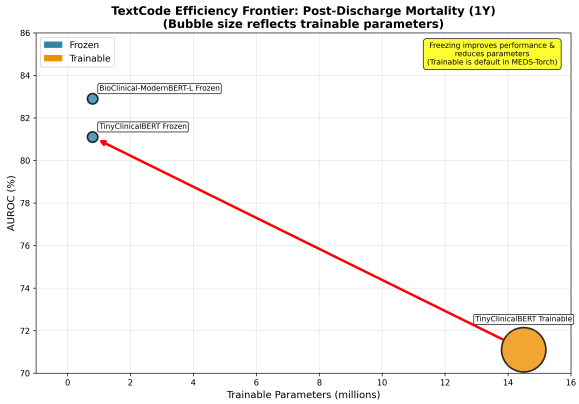

Figure 2: Efficiency frontier for TextCode variants on post-discharge mortality. Bubble size reflects trainable parameters. Freezing the encoder offers a superior performance/cost trade-off.

4. Conclusion

Our findings reveal that current EHR tokenization approaches may be over-engineered for certain clinical prediction tasks. The consistent lack of impact from explicit time features suggests that for transformer architectures applied to prediction tasks, these may provide limited additional signal beyond the inherent sequence order captured through positional embeddings. For mortality tasks, code-only models generally outperformed or matched models with code plus time embeddings, suggesting that time features without values may add noise rather than signal.

The strong performance of frozen encoders over trainable ones highlights a key trade-off. Fine-tuning entire LLMs risks catastrophic forgetting and overfitting, while requiring significant computational resources. In contrast, frozen encoders act as powerful, regularized feature extractors, leveraging rich pretrained knowledge while only training small projection layers. This approach proved more stable and effective across all evaluated tasks.

Our focus on Triplet and TextCode was motivated by their contrasting initial performance (highest vs lowest win rate in MEDS-Torch benchmarks) and their structural similarity that enables clean ablations. Given that code information emerged as the most critical predictive signal, we hypothesized that TextCode's underperformance reflected unfair experimental conditions rather than fundamental lim-

Table 1: TextCode results on post-discharge mortality. BioC-L is BioClinical-ModernBERT-L. † Bonferroni-corrected; * uncorrected $p < 0.05$. Blank cells are not significant.

Axis	Baseline	Variant
Mapping	Original mapping: 71.2±0.8	Enhanced mapping: 71.1±0.3
Trainability	Tiny-ClinicalBERT (Train.): 71.1±0.3	Tiny-ClinicalBERT (Frozen): 81.1±1.3† ↑
Size	Tiny-ClinicalBERT (Frozen): 81.1±1.3	BioC-L (Frozen): 82.9±0.9† ↑
Domain	Clinical (BioC-L): 82.9±0.9	General (Qwen3): 81.5±1.2† ↓

itations. Our controlled comparison shows that frozen BioClinical-ModernBERT-Large substantially narrows TextCode's performance gap to Triplet, bringing it within approximately 1 AUROC point (compared to over 10 points in initial benchmarks), with Triplet maintaining a small advantage across tasks under our fixed experimental settings. EIC (Everything In Code), the third major tokenization paradigm in MEDS-Torch, discretizes all modalities into categorical tokens rather than using continuous encoders, requiring fundamentally different experimental designs beyond our current scope.

These findings suggest that simpler, more parameter-efficient approaches can achieve strong performance in clinical time series modeling, though optimal tokenization remains task-dependent. Our controlled evaluation framework enables fairer comparisons between tokenization methods. While these insights are derived from transformer-based models on MIMIC-IV for prediction tasks, they provide a foundation for broader investigation across architectures, datasets, and clinical applications. Future work should explore whether these patterns hold for other clinical tasks, model architectures (e.g., RNNs, state-space models) and across multiple institutions.

References

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. Tokenizer choice for llm training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, 2024.

Rafi Al Attrach, Pedro Moreira, Rajna Fani, Renato Umeton, and Leo Anthony Celi. Conversational llms simplify secure clinical data access, understanding, and analysis. *arXiv preprint arXiv:2507.01053*, 2025.

Xi Chen, Yateng Tang, Jiarong Xu, Jiawei Zhang, Siwei Zhang, Sijia Peng, Xuehao Zheng, and Yun Xiong. Rethinking time encoding via learnable transformation functions. *arXiv preprint arXiv:2505.00887*, 2025.

Andrew R. Garcia. Time2vec neural network components, 2024. URL https://github.com/andrewrgarcia/time2vec. Keras and PyTorch implementation of Time2Vec.

Stefan Hegselmann, Georg von Arnim, Tillmann Rheude, Noel Kronenberg, David Sontag, Gerhard Hindricks, Roland Eils, and Benjamin Wild. Large language models are powerful electronic health record encoders. *arXiv preprint arXiv:2502.17403*, 2025.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1): 1, 2023.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):7155, 2020.

Matthew McDermott, Haoran Zhang, Lasse Hansen, Giovanni Angelotti, and Jack Gallifant. A closer look at auroc and auprc under class imbalance. *Advances in Neural Information Processing Systems*, 37:44102–44163, 2024.

Matthew B. A. McDermott, Justin Xu, Teya S. Bergamaschi, Hyewon Jeong, Simon A. Lee, Nassim Oufattole, Patrick Rockenschaub, Kamile Stankeviciute, Ethan Steinberg, Jimeng Sun, Robin P. van de Water, Michael Wornow, John Wu, and Zhenbang Wu. Meds: Building models and tools in a reproducible health ai ecosystem. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6243–6244, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737608. URL https://doi.org/10.1145/3711896.3737608.

Medical Event Data Standard Organization. Mimic-iv meds etl pipeline, 2024. URL https://github.com/Medical-Event-Data-Standard/MIMIC_IV_MEDS. Software for extracting MIMIC-IV dataset into MEDS format.

Nassim Oufattole, Teya Bergamaschi, Pawel Renc, Aleksia Kolo, Matthew BA McDermott, and Collin Stultz. Meds-torch: An ml pipeline for inductive experiments for ehr medical foundation models. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

Chao Pang, Xinzhuo Jiang, Nishanth Parameshwar Pavinkurve, Krishna S Kalluri, Elise L Minto, Jason Patterson, Linying Zhang, George Hripcsak, Gamze Gürsoy, Noémie Elhadad, et al. Cehr-gpt: Generating electronic health records with chronological patient timelines. *arXiv preprint arXiv:2402.04400*, 2024.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86, 2021.

Pawel Renc, Michal K Grzeszczyk, Nassim Oufattole, Deirdre Goode, Yugang Jia, Szymon Bieganski, Matthew McDermott, Jaroslaw Was, Anthony E Samir, Jonathan W Cunningham, et al. Foundation model of electronic medical records for adaptive risk estimation. *arXiv preprint arXiv:2502.06124*, 2025.

Omid Rohanian, Mohammadmahdi Nouriborji, Hannah Jauncey, Samaneh Kouchaki, Farhad Nooralahzadeh, Lei Clifton, Laura Merson, David A Clifton, ISARIC Clinical Characterisation Group, et al. Lightweight transformers for clinical natural language processing. *Natural language engineering*, 30(5):887–914, 2024.

Thomas Sounack, Joshua Davis, Brigitte Durieux, Antoine Chaffin, Tom J Pollard, Eric Lehman, Alistair EW Johnson, Matthew McDermott, Tristan Naumann, and Charlotta Lindvall. Bioclinical modernbert: A state-of-the-art long-context encoder for biomedical and clinical nlp. *arXiv preprint arXiv:2506.10896*, 2025.

Xiaorui Su, Shvat Messica, Yepeng Huang, Ruth Johnson, Lukas Fesser, Shanghua Gao, Faryad Sahneh, and Marinka Zitnik. Multimodal medical code tokenizer. *arXiv preprint arXiv:2502.04397*, 2025.

Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

Shane Waxler, Paul Blazek, Davis White, Daniel Sneider, Kevin Chung, Mani Nagarathnam, Patrick Williams, Hank Voeller, Karen Wong, Matthew Swanhorst, et al. Generative medical event models improve with scale. *arXiv preprint arXiv:2508.12104*, 2025.

Michael Wornow, Rahul Thapa, Ethan Steinberg, Jason Fries, and Nigam Shah. Ehrshot: An ehr benchmark for few-shot evaluation of foundation models. *Advances in Neural Information Processing Systems*, 36:67125–67137, 2023.

Justin Xu, Jack Gallifant, Alistair EW Johnson, and Matthew McDermott. Aces: Automatic cohort extraction system for event-stream datasets. *arXiv preprint arXiv:2406.19653*, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Appendix A. Implementation Details

A.1. TextCode Encoder Architecture

Freezing Mechanism: The flexible TextCode encoder implements two distinct pathways. For frozen models, we precompute embeddings for all vocabulary codes using the pretrained language model, cache them as tensors, and disable gradient computation (`requires_grad=False`). During training, embeddings are retrieved via fast tensor indexing rather than forward passes through the language model. For trainable models, we perform on-the-fly BERT inference with gradients enabled, allowing end-to-end optimization.

Caching System: Frozen models utilize a sophisticated caching mechanism that precomputes embeddings for the entire vocabulary (∼1.2K codes) and stores them as pickle files. The cache maps vocabulary indices to native model dimensions (768 for Tiny-ClinicalBERT, 1024 for BioClinical-ModernBERT-large, 1024 for Qwen3), which are then projected to the token dimension (128). During training, unique codes are identified using `fast_unique_with_inverse`, embeddings are retrieved via tensor indexing, and a learned projection layer maps to the model's token dimension (128).

Enhanced Mapping: The original MEDS-Torch mapping suffered from 75% missing descriptions, using fallback codes that may not capture clinical semantics. Our enhanced mapping transforms structured code identifiers into human-readable descriptions (e.g., "INFUSION_END//225166" becomes "Infusion of potassium chloride ended"), providing richer textual descriptions for language model encoding. Without rich textual descriptions, language model-based tokenizers may not distinguish between semantically related codes or leverage their extensive pretraining on medical literature. Our enhanced mapping ensures that any observed performance differences between Triplet and TextCode approaches can be attributed to the fundamental tokenization strategy rather than to data preprocessing artifacts.

A.2. Model Specifications

- **Tiny-ClinicalBERT** (Rohanian et al., 2024) (nlpie/tiny-clinicalbert): 15M parameters, 768-dim embeddings
- **BioClinical-ModernBERT-Large** (Sounack et al., 2025) (thomas-sounack/BioClinical-ModernBERT-large): 396M parameters, 1024-dim embeddings
- **Qwen3-Embedding-0.6B** (Zhang et al., 2025) (Qwen/Qwen3-Embedding-0.6B): 596M parameters, 1024-dim embeddings

A.3. Controlled Experimental Settings

```
meds-torch-train experiment=triplet_mtr
  paths.data_dir=triplet_tensors
  paths.meds_cohort_dir=MEDS_cohort
  data.task_root_dir=MEDS_cohort/tasks
  trainer.accelerator=auto
  trainer.devices=1
  trainer.precision=32
  trainer.strategy=auto
  logger=csv
  ++model.token_dim=128
  ++data.dataloader.batch_size=64
  ++trainer.max_epochs=10
  ++data.dataloader.num_workers=6
```

A.4. Extended Time Encoder Results

- **Time2Vec:** We evaluated k ∈ {1, 2, 3, 4, 5, 6, 8, 10, 20, 50} for the sinusoidal component count using a standard PyTorch implementation (Garcia, 2024). Raw p-values showed occasional improvements (k=6,10 for in-hospital mortality AUROC), but none survived Bonferroni correction across the full experimental grid.
- **LeTE:** Learnable time encodings mixed Fourier basis functions with spline multilayer perceptrons. We tested balanced (equal Fourier/spline weights), spline-heavy, and Fourier-heavy configurations. All variants performed comparably to the baseline continuous value encoder, suggesting that simple linear time encoding suffices for these clinical prediction tasks.

A.5. Computational Efficiency

Frozen TextCode models achieve 15× reduction in trainable parameters (14.5 M → < 1 M) while improving performance. Training time decreases substantially due to cached embeddings eliminating repeated language model forward passes. Even with larger frozen models (396M ModernBERT-Large), the computational footprint during training remains minimal since only the projection layer requires gradient computation.

Appendix B. Complete Ablation Results

Complete numeric results for the Triplet component ablations across all four clinical tasks are provided in Table 2.

Appendix C. Full TextCode Results Across All Tasks

Full per-task results for the TextCode experiments are provided in Table 3.

Appendix D. Learnable Time Encoders

We evaluated Time2Vec and LeTE. For Time2Vec, we explored a range of values for the number of sinusoidal components, k. We began with small values and increased the step size after observing diminishing returns. No variant offered a statistically significant improvement over the simple linear baseline, as shown in Table 4.

Table 2: Triplet ablations (AUROC $\mu \pm \sigma$; N=10). † Bonferroni-corrected $p < 0.05$; * uncorrected. Direction arrows compare to Triplet.

Task	**Triplet**	**No time**	**No value**	**Code-only**
In-hospital mortality (24h)	82.3±0.7	82.0±0.9	79.5±0.7† ↓	80.2±0.8† ↓
ICU mortality (24h)	78.5±0.5	78.0±0.8	75.4±0.4† ↓	76.2±0.8† ↓
Post-discharge mortality (1y)	84.2±0.4	84.1±0.3	81.9±0.2† ↓	81.9±0.5† ↓
Readmission (30d)	69.9±0.8	69.8±0.3	69.5±0.4	68.8±1.0* ↓

Table 3: Full TextCode results across all tasks (AUROC $\mu \pm \sigma$; N=10).

Variant	**In-hosp. mort.**	**ICU mort.**	**Post-disch. mort.**	**Readmission**
Tiny-ClinicalBERT (Train.)	69.5±1.3	64.3±0.8	71.1±0.3	61.1±0.4
Tiny-ClinicalBERT (Frozen)	72.5±1.6	72.8±0.9	81.1±1.3	65.6±1.1
BioClinical-ModernBERT-Large (Frozen)	76.2±1.9	75.6±1.3	82.9±0.9	69.5±1.7
Qwen3 (Frozen)	75.1±1.2	72.8±0.9	81.5±1.2	67.2±1.2

Table 4: Best learnable time encoding per task vs baseline (AUROC $\mu \pm \sigma$; N=10).

Task	**Baseline (CVE)**	**Best Variant**	**Wilcoxon p**
In-hospital mortality (24h)	82.3±0.7	T2V(k=6): 82.7±0.8	0.193
ICU mortality (24h)	78.5±0.5	Spline (Heavy): 78.9±0.4	0.037
Post-discharge mortality (1y)	84.2±0.4	Spline (Bal.): 84.8±0.8	0.027
Readmission (30d)	69.9±0.8	T2V(k=2): 70.3±0.7	0.037