CACHED FOUNDATION MODEL SUMMARIES FOR MEMORY-EFFICIENT CLINICAL TIME SERIES INFERENCE

Rafi Al Attrach[1,2] **Rajna Fani**[1,2] **David Restrepo**[3]
Yugang Jia[1] **Leo Anthony Celi**[1,4,5†] **Peter Schüffler**[2,6,7†]

[1]Massachusetts Institute of Technology (MIT), USA
[2]CIT, Technical University of Munich (TUM), Germany
[3]MICS, CentraleSupélec – Université Paris-Saclay, France
[4]Harvard Medical School, USA
[5]Beth Israel Deaconess Medical Center, USA
[6]Institute of Pathology, TUM, Germany
[7]Munich Center for Machine Learning (MCML), Germany

{rafiaa, rajnaf, yugang, lceli}@mit.edu
david.restrepo@centralesupelec.fr peter.schueffler@tum.de
[†]Shared corresponding authors.

ABSTRACT

Transformer-based models for clinical time series face a deployment bottleneck: patient histories can span thousands of irregularly spaced events, yet inference hardware imposes strict memory budgets. We study a simple decoupling strategy in which a pretrained foundation model compresses a patient's historical events into a fixed-size cached summary offline, and a lightweight prediction model processes only a short window of recent events conditioned on that summary at inference time. Through 252 experiments on MIMIC-IV we characterize *when* this strategy is worthwhile. The central finding is a clear pattern of diminishing returns: cached summaries yield a 6.5% relative AUROC gain when the recent window is limited to 8 events ($p < 0.001$), but the benefit shrinks to a statistically insignificant 0.1% once the window reaches 256 events. We further show that modulating event representations with the summary (FiLM) outperforms treating it as an additional input token ($p < 0.001$), and that summaries of recent history are more informative than those of distant history ($p < 0.01$). Together, these results provide actionable guidance for allocating context budgets when deploying sequence models on long, irregular time series under memory constraints.

Track: Research

1 INTRODUCTION

Clinical event sequences recorded in electronic health records form irregular time series with distinctive properties: events arrive at non-uniform intervals, each event combines a categorical code with an optional numeric value, and individual patient histories can span thousands of observations over months or years. Transformer-based models have achieved strong results on clinical prediction tasks (Li et al., 2020; Rasmy et al., 2021), but self-attention's quadratic memory complexity creates a gap between training and deployment. A model trained on 512-event windows may exceed the memory available on inference hardware when applied to patients with longer records.

Several lines of work address the long-sequence challenge at the architecture level. Sparse attention patterns (Beltagy et al., 2020; Zaheer et al., 2020) reduce complexity to linear. State-space models (Gu & Dao, 2023; Fallahpour et al., 2024) bypass the attention bottleneck entirely. Hardware-level optimizations such as Flash Attention (Dao et al., 2022) improve throughput without changing

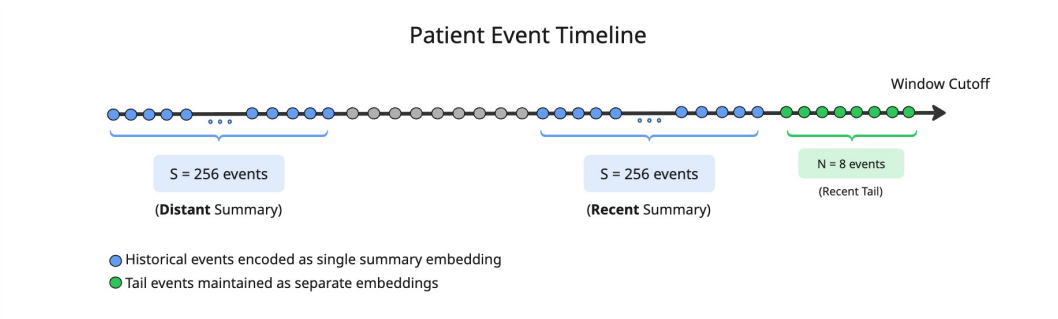

Figure 1: Deployment architecture. Historical events (blue) are encoded offline by a clinical language model into a cached summary vector. At inference, the prediction model processes only the recent tail (green) conditioned on the cached summary. The summary can represent either distant history or the period immediately preceding the tail.

the asymptotic scaling. These approaches require practitioners to adopt new architectures or specialized hardware.

We investigate a complementary strategy that operates at the *input* level rather than the model level. A clinical language model (BioClinical-ModernBERT; Sounack et al. 2025) encodes each patient's past events into a fixed-size vector during offline preprocessing. The inference model then processes only a short window of recent events, conditioned on this cached summary (Figure 1). The idea of compressing patient histories into dense vectors has precedent (Choi et al., 2016a;b), but what remains unclear is *when* such compression helps relative to simply extending the recent window.

We frame this as a context budget allocation problem. Given fixed inference memory, a practitioner can process N recent events directly or process fewer recent events supplemented by a cached summary of S historical events. Which allocation yields better predictions? We study this through 252 experiments on MIMIC-IV (Johnson et al., 2023), varying $N \in \{8, 16, 32, 64, 128, 256\}$, $S \in \{128, 256, 512\}$, the integration mechanism, and the temporal source of the summary.

Our main findings are: (1) cached summaries provide meaningful improvement only when the recent window is short ($N \leq 32$), with gains diminishing to statistical insignificance at $N \geq 128$; (2) FiLM conditioning (Perez et al., 2018) outperforms token injection for integrating summaries; and (3) summaries from recent history outperform those from distant history. These results offer deployment guidance and speak to a general question in time series modeling: how to balance high-resolution recent observations against compressed representations of longer history.

2 Methods

2.1 Data and task

We use MIMIC-IV (Johnson et al., 2023), a public critical care database, processed into the Medical Event Data Standard (MEDS) format (McDermott et al., 2025) via the MIMIC-IV MEDS ETL pipeline (McDermott & Xu, 2025). Each clinical event is a triplet: a medical code, a time delta (days since the previous event), and an optional numeric value. For instance, a glucose measurement becomes (`LAB//220621//mg/dL`, Δt=0.08, 120). We focus on predicting in-ICU mortality within 24 hours, selected for its long event sequences (median 7,351 events per patient across 35,550 patients; see Appendix E). All experiments use the MEDS-Torch framework (Oufattole et al., 2024).

2.2 Architectures

We compare three architectures built on a standard Transformer encoder (Vaswani et al., 2017) (details in Appendix C).

Baseline. Processes only the N most recent events. Memory scales with N.

Oracle. Processes the full sequence of $N + S$ events with standard attention. Memory scales with total sequence length.

Cached Summary. Processes N recent events plus a precomputed summary vector of S historical events. Memory scales with N only, since the history is compressed to a fixed-size vector before inference.

2.3 Summary generation and integration

Historical summaries are generated offline. We extract the S historical events for each patient, convert medical codes to text descriptions (see Appendix L), and encode the concatenated text with BioClinical-ModernBERT (Sounack et al., 2025), a clinical language model supporting sequences up to 8,192 tokens. The final hidden state yields a 768-dimensional summary vector, cached to disk at 5.35 KB per patient (Appendix D).

We compare two integration mechanisms:

Token Injection. The summary vector replaces the code embedding at the first sequence position. It influences other tokens only through self-attention.

Feature-wise Linear Modulation (FiLM). The summary $\mathbf{z}_s \in \mathbb{R}^{768}$ generates scale and shift parameters that modulate all code embeddings:

$$\gamma = W_\gamma \mathbf{z}_s + b_\gamma \tag{1}$$

$$\beta = W_\beta \mathbf{z}_s + b_\beta \tag{2}$$

$$\mathbf{e}'_{\text{code}} = \gamma \odot \mathbf{e}_{\text{code}} + \beta \tag{3}$$

where $W_\gamma, W_\beta \in \mathbb{R}^{d \times 768}$ are learned projections (Perez et al., 2018). This allows historical context to modulate how recent events are represented before self-attention.

2.4 Experimental protocol

We conducted a grid search over $N \in \{8, 16, 32, 64, 128, 256\}$ and $S \in \{128, 256, 512\}$, comparing summaries from **recent** history (immediately preceding the tail) and **distant** history (earliest events in the record), as shown in Figure 1. Each configuration was run with three random seeds. We report AUROC as the primary metric and assess statistical significance with paired Wilcoxon signed-rank tests.

3 Results

We conducted 252 experiments spanning 6 values of N, 3 values of S, 2 integration methods, 2 summary sources, and 3 random seeds.

3.1 Diminishing returns of cached summaries

Figure 2 shows the central finding: the relative AUROC improvement from cached summaries decays as the recent context window grows. At N=8, summaries provide a 6.5% relative improvement ($p < 0.001$). This drops steadily to 0.1% at N=256 (not significant). The pattern holds across all summary sizes tested.

The improvement remains statistically significant through N=64 ($p < 0.05$). At $N \geq 128$, improvements are not significant, indicating that cached summaries do not reliably help at these context lengths. The intuition is straightforward: when the recent window is short, the model lacks access to predictive events, and the compressed summary recovers some of this missing signal. As the window grows, the model sees these events directly, making compressed history redundant.

3.2 Integration method and summary source

Table 1 summarizes the ablation results. FiLM conditioning outperforms token injection ($p < 0.001$), with a mean AUROC improvement of 0.88 percentage points across all configurations.

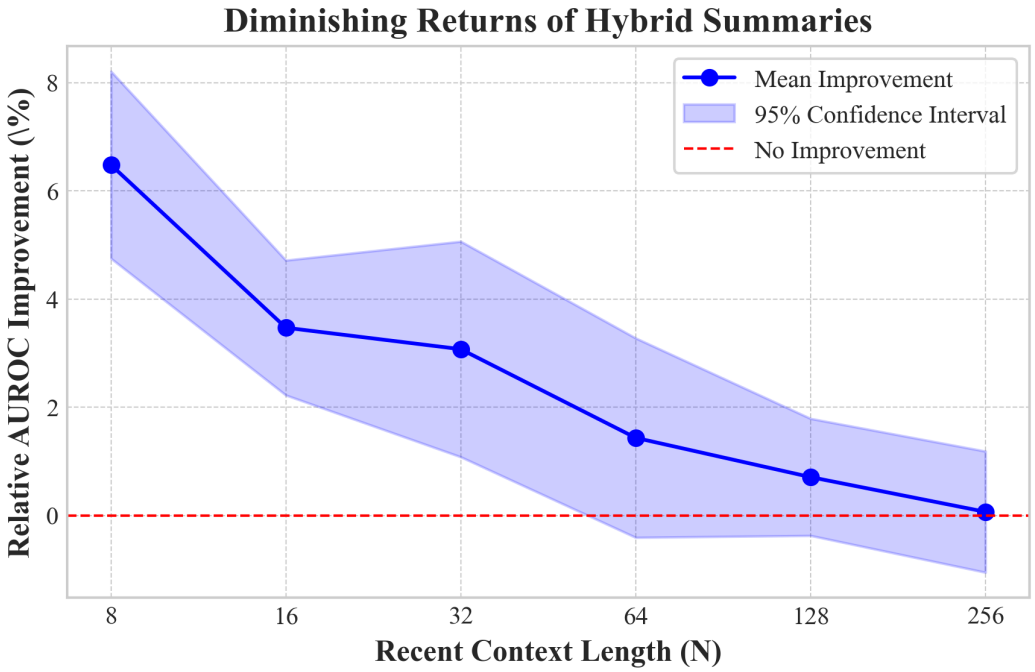

Figure 2: Relative AUROC improvement of the cached summary architecture (FiLM, S=256) over the baseline, by recent context length N. Error bars show 95% confidence intervals across three seeds.

Table 1: Ablation results averaged across all N and S configurations. p-values from paired Wilcoxon signed-rank tests.

Method	**Mean AUROC**
Integration Method ($p < 0.001$)	
FiLM	**0.7542**
Token Injection	0.7454
Summary Source ($p < 0.01$)	
Recent History	**0.7542**
Distant History	0.7453

The advantage of FiLM reflects a structural difference in how historical context enters the model. Token injection treats the summary as another sequence element competing for attention. FiLM instead uses the summary to modulate the *representation* of recent events, allowing history to serve as interpretive context rather than an additional observation. This maps naturally to clinical reasoning: a glucose reading of 180 mg/dL carries different meaning for a patient with diabetes than one without metabolic history.

Recent summaries (the S events immediately before the tail) outperform distant summaries (the first S events in the record) with $p < 0.01$. For acute care prediction, events closer in time carry more predictive signal than early historical events. Extended results, per-configuration breakdowns, and effect sizes appear in Appendices G through I.

4 Conclusion

We evaluated a deployment architecture that separates historical encoding (offline, via a clinical language model) from inference (online, memory-efficient) for clinical event sequences. Across 252 experiments on MIMIC-IV, three patterns emerge. First, cached summaries help when the recent

window is short ($N \leq 32$) and become negligible at longer windows ($N \geq 128$). Second, FiLM integration, which uses history to condition the representation of recent events, outperforms token injection. Third, recent history is more informative than distant history for acute care prediction.

For practitioners deploying sequence models under memory constraints, these results suggest a straightforward heuristic: when hardware limits the input length to a few dozen events, supplement with cached summaries using FiLM conditioning; when longer windows are feasible, skip the summarization overhead entirely. Our evaluation focuses on one prediction task (24-hour ICU mortality) on one dataset (MIMIC-IV) with one summarization model; we discuss limitations and extensions in Appendix P.

REFERENCES

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.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pp. 301–318. PMLR, 2016a.

Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Medical concept representation learning from electronic health records and its application on heart failure prediction. *arXiv preprint arXiv:1602.03686*, 2016b.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Adibvafa Fallahpour, Mahshid Alinoori, Wenqian Ye, Xu Cao, Arash Afkanpour, and Amrit Krishnan. Ehrmamba: Towards generalizable and scalable foundation models for electronic health records. *arXiv preprint arXiv:2405.14567*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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.

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 and Justin Xu. MIMIC-IV MEDS: An ETL pipeline to extract MIMIC-IV data into the MEDS format, May 2025. URL `https://github.com/Medical-Event-Data-Standard/MIMIC_IV_MEDS`.

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, pp. 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`.

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.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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.

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.

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.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

A Data and code availability

For reproducibility, we detail the data and code used in this study. We use the publicly available MIMIC-IV v3.1 dataset (Johnson et al., 2023), processed into the Medical Event Data Standard (MEDS) format (McDermott et al., 2025) via the MIMIC-IV MEDS ETL pipeline (McDermott & Xu, 2025). Experiments are implemented using the MEDS-Torch framework (Oufattole et al., 2024). The code is available at `https://github.com/rafiattrach/cached-summaries-ehr-inference`.

B Ethics statement

This research uses the publicly available, de-identified MIMIC-IV dataset. Access requires credentialed authorization through PhysioNet. The dataset has been stripped of protected health information in accordance with HIPAA requirements. No additional IRB approval was required for this secondary analysis of de-identified data.

C Technical details

Hardware. Experiments were conducted on AWS `ml.g5.4xlarge` instances with NVIDIA A10G GPUs (24 GB VRAM).

Model architecture. The Transformer encoder uses 2 layers, 2 attention heads, and 128-dimensional token embeddings. This architecture follows MEDS-Torch defaults for the triplet encoder experiments.

Training configuration. All models were trained for up to 10 epochs with early stopping (patience=3). We used the Adam optimizer with learning rate 10^{-3}, batch size 64, and 32-bit precision. We used a patient-level 80/10/10 split (`train`, `tuning`, `held_out`) from the respective cohort.

FiLM implementation. The FiLM layer uses two linear projections $W_\gamma, W_\beta \in \mathbb{R}^{128\times768}$ to map the 768-dimensional summary to 128-dimensional scale (γ) and shift (β) vectors:

$$\gamma = W_\gamma \mathbf{z}_s + b_\gamma \in \mathbb{R}^{128} \tag{4}$$

$$\beta = W_\beta \mathbf{z}_s + b_\beta \in \mathbb{R}^{128} \tag{5}$$

$$\mathbf{e}'_{\text{code}} = \gamma \odot \mathbf{e}_{\text{code}} + \beta \tag{6}$$

These parameters are learned jointly with the rest of the model.

D PREPROCESSING COST DETAILS

Summarization model. We use BioClinical-ModernBERT-base (Sounack et al., 2025) (hidden size 768, 22 layers, 12 attention heads; 150M parameters) as a long-context, domain-adapted clinical encoder to generate fixed-size cached summaries.

Code-to-text mapping. Medical codes are converted to human-readable descriptions before encoding. To address incomplete coverage in the original MEDS vocabulary mappings, we used the MIMIC-IV MCP Server (Attrach et al., 2025) to generate comprehensive descriptions for all codes. This increased mapping coverage from 25% to 100%, ensuring that all medical codes have meaningful textual descriptions rather than fallback identifiers.

Per-sample timing. On a single A10G GPU:

- Model forward pass: 20.3 ms (median)
- Tokenization: 1.5 ms (median)
- Total per patient: 21.9 ms (median)
- Cold start overhead: 2.6 seconds (first sample only)

Cohort-level timing. For our cohort of 35,550 patients:

- Wall-clock time per configuration: 72.5 minutes
- Total for all 36 configurations tested: 43.5 GPU-hours

Storage. Each cached summary file contains:

- 768-dimensional embedding (float32): 3.0 KB
- Metadata (patient ID, window indices, configuration): 2.35 KB
- Total per patient: 5.35 KB
- Full cohort per configuration: 190 MB

Amortization. The 252 training experiments in this paper reused cached summaries generated during preprocessing. Since each configuration's summaries (72.5 minutes to generate) were reused across multiple experiments, the effective preprocessing overhead per training run was $72.5/252 \approx 0.29$ minutes, adding approximately 2.5% to the average training time of 11.6 minutes.

Table 2: Preprocessing cost summary.

Metric	Value
Time per patient	22 ms
Time per cohort configuration	72.5 min
Storage per patient	5.35 KB
Storage per cohort	190 MB
Amortized cost (252 experiments)	0.29 min/exp

E TASK STATISTICS

Table 3 shows event sequence statistics for MEDS-formatted MIMIC-IV. We selected 24-hour ICU mortality due to its long sequences, which makes it well-suited for evaluating long-context strategies.

Table 3: Event sequence lengths in MEDS-formatted MIMIC-IV.

Task	Patients	Med. Events
ICU mort. (24h)	**35,550**	**7,351**
Hosp. mort. (24h)	164,900	985
Readmission (30d)	145,004	814
Post-disch. mort. (1y)	210,022	541

F Extended discussion

F.1 Practical guidelines

Our results quantify the trade-off between high-resolution recent context (the N most recent events) and compressed historical context (a summary of S prior events):

- **When inference hardware limits** $N \leq 32$: Cached summaries provide 2–6% relative AUROC improvement. The preprocessing cost (72 min per cohort configuration) and storage (190 MB) are justified by the meaningful performance gains.
- **When** $N \geq 128$ **is feasible**: Cached summaries add less than 1% improvement (not statistically significant at N=256). The preprocessing overhead is not justified; resources are better spent increasing N directly.
- **Integration choice**: Use FiLM over token injection. The FiLM approach allows historical context to modulate how recent events are interpreted, rather than competing with them for attention.

F.2 Implications for model design

The superiority of FiLM over token injection has broader implications for clinical sequence model design. Token injection forces historical context to compete with recent events for attention, treating the summary as just another sequence element. FiLM instead allows historical context to *condition* how recent events are interpreted, a fundamentally different computational role.

This distinction maps to clinical reasoning. When a physician reviews a patient's current lab results, they do so in the context of the patient's history. The history does not compete for attention with current observations; rather, it provides interpretive context. A creatinine level of 2.0 mg/dL has different implications depending on whether the patient has chronic kidney disease. FiLM captures this contextual interpretation by modulating the representation of current events based on historical context.

F.3 Deployment scenarios

The cached summary architecture is most appropriate for specific deployment scenarios:

Edge devices and resource-constrained environments. Hospitals deploying AI at the point of care may face strict memory constraints. A clinical workstation or embedded device may not have the GPU memory to process full patient histories. The cached summary architecture enables deployment on such devices: the expensive summarization happens once (potentially in the cloud), and the lightweight inference model processes only recent events plus a cached vector.

High-throughput serving. Clinical decision support systems serving many concurrent requests benefit from predictable, low-latency inference. Processing 32 tokens plus a cached summary is faster and more memory-efficient than processing 512 tokens, enabling higher throughput with the same hardware.

Multi-task systems. When multiple prediction tasks (mortality, readmission, length of stay) share the same patient cohort, the preprocessing investment is amortized across all tasks. Summaries are computed once per patient and reused, making the per-task overhead negligible.

G EXTENDED RESULTS

This section provides detailed performance metrics and computational resource analysis across all experimental configurations. We report results aggregated over three random seeds with standard deviations to characterize variability.

G.1 PERFORMANCE METRICS BY ARCHITECTURE

Table 4 summarizes the primary evaluation metrics for each architecture type. The cached summary architecture with FiLM integration achieves intermediate performance between the baseline (which sees only recent events) and the oracle (which processes the full sequence). The lower variance of the oracle reflects the stability that comes from having access to more information.

Table 4: Performance metrics (mean $\pm$ std across seeds). Cached results use recent summaries with S=256.

Architecture	AUROC	AUPRC	Accuracy	Loss
Baseline	0.738 ± 0.051	0.229 ± 0.048	0.914 ± 0.001	0.234 ± 0.045
Cached (Token)	0.745 ± 0.042	0.226 ± 0.047	0.914 ± 0.001	0.231 ± 0.044
Cached (FiLM)	0.754 ± 0.037	0.231 ± 0.044	0.914 ± 0.001	0.228 ± 0.042
Oracle	0.804 ± 0.007	0.267 ± 0.012	0.917 ± 0.001	0.198 ± 0.008

G.2 INTERACTION BETWEEN SUMMARY SIZE AND CONTEXT LENGTH

Figure 3 shows the interaction between summary size and recent context length. Larger summaries provide meaningful gains only when recent context is limited. At N=8, increasing S from 128 to 512 provides visible performance gains. At N=256, all three summary sizes perform nearly identically, confirming that larger summaries are not beneficial when recent context is already sufficient.

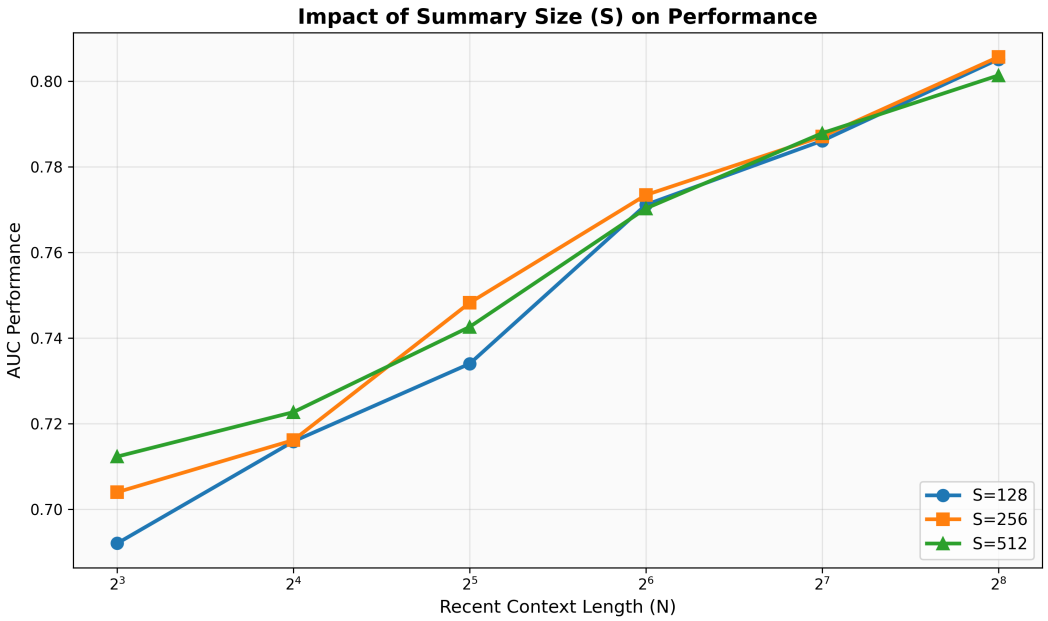

Figure 3: Impact of summary size (S) across recent context lengths (N). The benefit of larger summaries diminishes as recent context increases.

G.3 Absolute performance comparison

Figure 4 shows absolute AUROC values for the baseline, cached summary (FiLM), and oracle architectures. The cached summary architecture partially closes the gap between the baseline and the oracle, most effectively at small N. The oracle represents an upper bound: processing the full $N+S$ sequence with full attention.

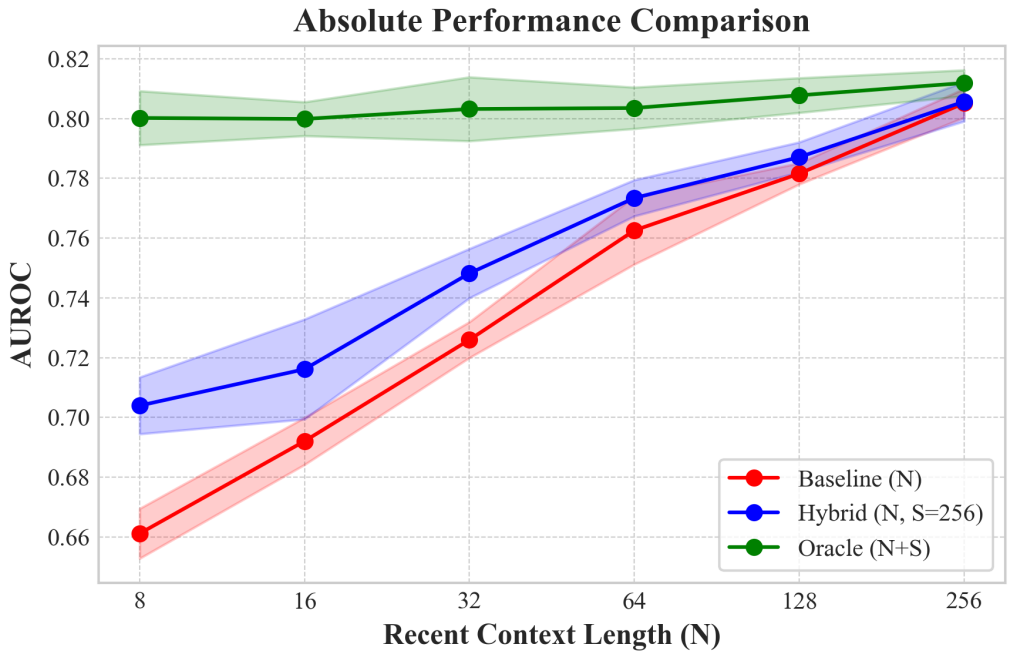

Figure 4: Absolute AUROC comparison. The cached summary architecture (FiLM, S=256) sits between baseline and oracle, with the gap narrowing as N increases.

Notably, the oracle's performance remains relatively stable across N values, indicating that once sufficient context is available, the model extracts consistent predictive signal. The baseline and cached summary architectures converge toward the oracle as N increases, suggesting that recent events carry the majority of predictive information for this task.

G.4 FiLM versus token injection across context lengths

Figure 5 shows the FiLM advantage across context lengths. The performance gap between FiLM and token injection is relatively consistent across N values, suggesting that the modulatory mechanism provides benefit regardless of how much recent context is available.

G.5 Computational resource utilization

Table 5 provides a breakdown of GPU memory and training time across architectures. Memory measurements were taken at peak allocation during forward passes. The cached summary architectures require additional training time due to the overhead of loading and processing the summary embeddings at each batch, though this overhead is modest compared to the memory savings relative to the oracle.

H Per-configuration results

This section provides granular results for each combination of recent context length (N) and summary size (S).

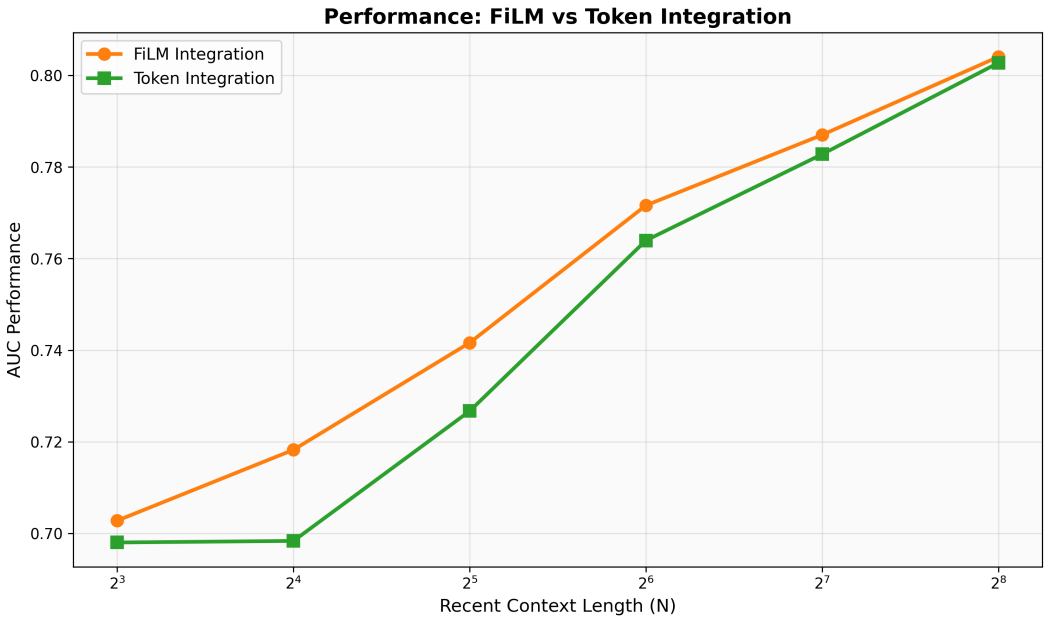

Figure 5: FiLM versus Token integration across recent context lengths. FiLM provides consistent improvement across all N values.

Table 5: Computational resource utilization (mean $\pm$ std across configurations).

Architecture	**GPU Memory**	**Training Time**
Baseline	591 $\pm$ 243 MB	3.3 $\pm$ 0.7 min
Cached (Token)	593 $\pm$ 243 MB	11.4 $\pm$ 2.5 min
Cached (FiLM)	594 $\pm$ 243 MB	11.6 $\pm$ 2.7 min
Oracle	1,292 $\pm$ 249 MB	3.6 $\pm$ 1.2 min

H.1 Relative improvement analysis

Table 6 quantifies the percentage improvement of the cached summary architecture over the baseline (FiLM, S=256). At N=8, summaries provide a 6.5% relative improvement ($p < 0.001$); this drops to 0.1% at N=256 (not significant).

Table 6: Relative AUROC improvement (%) over baseline (FiLM, S=256). Statistical significance from paired Wilcoxon tests.

N	**Rel. Improvement**	**Significance**
8	6.5%	$p < 0.001$
16	4.2%	$p < 0.01$
32	2.1%	$p < 0.05$
64	1.0%	$p < 0.05$
128	0.5%	n.s.
256	0.1%	n.s.

H.2 Integration method comparison

FiLM consistently outperforms token injection across all N values ($p < 0.001$). The aggregate comparison shows FiLM achieving mean AUROC of 0.7542 versus 0.7454 for token injection, representing a 0.88 percentage point improvement. This advantage is maintained across configurations, though the absolute magnitude of the summary benefit decreases with N for both integration methods.

I Statistical analysis details

I.1 Hypothesis testing framework

We use paired Wilcoxon signed-rank tests to compare configurations, as this non-parametric test does not assume normality of performance differences. For each comparison, we pair results by random seed and compute the test statistic over the three seeds.

Given the multiple comparisons in our study, we report both raw p-values and Bonferroni-corrected thresholds where applicable. For our primary claims (FiLM vs. token, recent vs. distant), we use a significance threshold of $\alpha = 0.01$ after correction.

I.2 Confidence interval construction

The 95% confidence intervals in Figure 2 are computed as:

$$\text{CI}_{95\%} = \bar{x} \pm 1.96 \times \frac{s}{\sqrt{n}} \tag{7}$$

where $\bar{x}$ is the sample mean, s is the sample standard deviation, and $n = 3$ is the number of seeds. We acknowledge that with only three seeds, these intervals rely on asymptotic normality assumptions.

I.3 Effect size analysis

Table 7 reports Cohen's d for key comparisons. Effect sizes are categorized as small ($d < 0.2$), medium ($0.2 \leq d < 0.8$), or large ($d \geq 0.8$).

The large effect size at N=8 and small effect size at N=256 quantifies the diminishing returns pattern described in the main text.

Table 7: Effect sizes (Cohen's d) for primary comparisons.

Comparison	Cohen's d	Size
FiLM vs. Token (all N)	0.42	Medium
Recent vs. Distant (all N)	0.38	Medium
Cached vs. Baseline (N=8)	1.12	Large
Cached vs. Baseline (N=256)	0.18	Small

J Ablation study details

J.1 Summary source: recent vs. distant

The "recent" summary covers the S events immediately preceding the N tail events. The "distant" summary covers the first S events in the patient's record. Across all configurations, recent summaries outperform distant summaries ($p < 0.01$), with a mean AUROC difference of approximately 0.9 percentage points.

The advantage of recent summaries aligns with clinical intuition: events closer in time to the prediction window are more likely to reflect the patient's current state. The performance gap narrows at larger N because the tail itself captures more recent context.

J.2 Summary size sensitivity

We evaluated summary sizes $S \in \{128, 256, 512\}$. S=256 provided the best balance across configurations, though differences between S=256 and S=512 were not statistically significant. This suggests that the summarization model may saturate in its ability to compress additional events, or that events beyond 256 provide marginal additional signal for this prediction task.

K Memory analysis

Transformer self-attention has $O(n^2)$ memory complexity in sequence length n. Table 8 shows measured GPU memory across sequence lengths for our architecture.

Table 8: GPU memory (MB) by recent context length (N). Oracle processes $N + S$ events (with S=256). Cached includes summary loading overhead. "Oracle Total" shows the total sequence length processed by Oracle.

N	Baseline	Cached	Oracle	Oracle Total
8	312	315	1,089	264
16	348	351	1,124	272
32	421	424	1,198	288
64	567	570	1,345	320
128	859	862	1,638	384
256	1,443	1,446	2,614	512

The cached architecture maintains memory consumption comparable to the baseline (within 3 MB overhead for summary embeddings), while the oracle requires roughly 2× the memory due to processing longer sequences. Memory scales approximately with $O(N^2)$ due to self-attention, though the exact relationship depends on batch size and model configuration.

L Code-to-text mapping details

L.1 Mapping coverage

The MEDS vocabulary contains 7,351 unique medical codes in our MIMIC-IV cohort. The default MEDS code descriptions provided coverage for approximately 25% of codes, with the remainder mapped to generic identifiers.

To address this, we used the MIMIC-IV MCP Server (Attrach et al., 2025) to retrieve human-readable labels directly from the MIMIC-IV database, increasing description coverage to 100%. This ensures that all medical codes have meaningful textual representations for the summarization model.

L.2 EXAMPLE MAPPINGS

To illustrate the scope of the enhanced mapping process, Table 9 shows representative examples across common event categories. In our cohort, the enhanced mapping replaced sparse identifiers for previously unmapped codes and achieved 100% description coverage (compared to approximately 25% coverage under the default mappings).

Table 9: Representative examples from the enhanced mapping showing the transformation from structured codes to clinical descriptions.

Original Code	Enhanced Description
Demographics and Baseline Factors	
`GENDER//F`	Gender recorded as female
`GENDER//M`	Gender recorded as male
`LAB//226512//kg`	Body weight (kg)
`LAB//226707//Inch`	Body height (Inch)
`BMI`	Body mass index (BMI)
Diagnoses (ICD-9 and ICD-10)	
`DIAGNOSIS//ICD//10//E119`	Type 2 diabetes mellitus without complications (ICD-10: E119)
`DIAGNOSIS//ICD//10//Z87891`	Personal history of nicotine dependence (ICD-10: Z87891)
`DIAGNOSIS//ICD//9//4019`	Unspecified essential hypertension (ICD-9: 4019)
Medications	
`MEDICATION//Acetaminophen//Administered`	Acetaminophen administered
`MEDICATION//START//Furosemide`	Furosemide started
`MEDICATION//Heparin//Stopped`	Heparin stopped
Infusions	
`INFUSION_START//221794`	Infusion of furosemide (Lasix) started
`INFUSION_END//225166`	Infusion of potassium chloride ended
`INFUSION_START//222168`	Infusion of propofol started
Procedures	
`PROCEDURE//START//227194`	Removal of endotracheal tube started
`PROCEDURE//END//225459`	Plain chest X-ray ended
`PROCEDURE//START//224263`	Central venous cannula insertion started
Laboratory Values	
`LAB//220621//mg/dL`	Glucose [Mass/volume] in Serum or Plasma (mg/dL)
`LAB//220615//mg/dL`	Creatinine [Mass/volume] in Serum or Plasma (mg/dL)
`LAB//220050//mmHg`	Systolic blood pressure (mmHg)
Administrative Events	
`HOSPITAL_ADMISSION//URGENT//PHYSICIAN REFERRAL`	Admitted urgently via physician referral
`ICU_ADMISSION//Medical Intensive Care Unit (MICU)`	Admitted to Medical Intensive Care Unit (MICU)
`TRANSFER_TO//transfer//Medicine`	Transferred to Medicine

M HYPERPARAMETER SENSITIVITY

M.1 LEARNING RATE

We evaluated learning rates in $\{10^{-4}, 10^{-3}, 10^{-2}\}$. The default 10^{-3} provided stable convergence across all configurations. Higher rates (10^{-2}) led to training instability in some configurations, while lower rates (10^{-4}) converged more slowly without improving final performance.

M.2 BATCH SIZE

We evaluated batch sizes in $\{32, 64, 128\}$. Batch size 64 was selected as the default, balancing memory efficiency and gradient noise. Larger batches (128) showed marginal improvements ($<$0.5% AUROC) but required more memory.

M.3 EARLY STOPPING PATIENCE

We used patience=3 epochs based on validation AUROC. Experiments with patience=5 showed no improvement in final test performance but increased training time by 20% on average due to additional epochs before convergence.

N REPRODUCIBILITY CHECKLIST

N.1 COMPUTE RESOURCES

- Hardware: AWS `ml.g5.4xlarge` (NVIDIA A10G, 24GB VRAM).
- Summary precomputation: 36 configurations; approximately 70–75 minutes per configuration on a single A10G; approximately 40–45 instance-hours in total.
- Model training: 252 training runs on a single A10G. Wall-clock time per run varies by model variant and context length, so we report training time per run rather than summarizing with a single aggregate GPU-hour estimate.
- Estimated cost (order of magnitude): on-demand `ml.g5.4xlarge` pricing varies by region and time; at roughly $2/hour, the precomputation cost is on the order of $100.

N.2 SOFTWARE VERSIONS

- Python: 3.11.11
- PyTorch: 2.7.0
- PyTorch Lightning: 2.5.1
- MEDS-Torch: 0.0.8
- Transformers: 4.55.2
- NumPy: 1.26.4

N.3 DATA AVAILABILITY

- MIMIC-IV v3.1: Available via PhysioNet (credentialed access required)
- MEDS preprocessing scripts: Available via anonymized repository (see Appendix A)
- Pretrained model: BioClinical-ModernBERT-base from HuggingFace

N.4 RANDOM SEEDS

All experiments were run with seeds $\{0, 1, 2\}$. Seeds were set using PyTorch Lightning's seed utility, which configures PyTorch, NumPy, and Python's random module for reproducibility.

O Related work

O.1 Long-context Transformers

Several architectural modifications address the quadratic complexity of self-attention. Longformer (Beltagy et al., 2020) combines local windowed attention with global attention on selected tokens. BigBird (Zaheer et al., 2020) adds random attention patterns. These approaches reduce complexity to $O(n)$ but require architectural changes.

Our work is complementary: rather than modifying the attention mechanism, we reduce the effective sequence length seen at inference time by compressing history into a dense vector. This separation allows practitioners to use standard Transformer implementations while achieving memory efficiency.

O.2 State-space models for clinical data

Mamba (Gu & Dao, 2023) and similar state-space models achieve linear complexity without attention modification. EHR-Mamba (Fallahpour et al., 2024) applies this architecture to clinical sequences. These models show promise but require practitioners to adopt new architectures and may not match Transformer performance on all tasks.

Our approach maintains compatibility with the extensive ecosystem of pretrained clinical Transformers while addressing deployment constraints.

O.3 Patient history summarization

Prior work has explored compressing patient histories into dense representations. Doctor AI (Choi et al., 2016a) uses recurrent networks to encode visit sequences. Med2Vec (Choi et al., 2016b) learns code embeddings through skip-gram objectives. These approaches predate the Transformer era.

Our contribution is characterizing *when* such compression provides value in modern Transformer-based pipelines, providing practitioners with guidance on when to invest in summarization infrastructure versus simply processing longer sequences.

O.4 Feature-wise modulation in healthcare

FiLM conditioning (Perez et al., 2018) was originally proposed for visual question answering. Its application to clinical prediction is less explored. Our results suggest that modulation-based integration of auxiliary information (historical summaries) outperforms concatenation-based approaches (token injection) in the EHR domain.

This finding may generalize to other settings where auxiliary context (patient demographics, prior diagnoses, clinical notes) must be integrated with sequential event data.

P Failure cases and limitations

P.1 Where cached summaries do not help

Analysis of individual predictions reveals cases where cached summaries do not improve performance:

Patients with short histories. For patients with fewer than $N + S$ events, the summary contains the same information as the tail. In these cases, cached summaries provide no additional signal.

Acute events in the tail. When mortality is determined by acute events in the recent window (e.g., cardiac arrest, massive hemorrhage), historical context is less relevant. The baseline and cached summary architectures perform similarly on these cases.

Misaligned summary content. If the summary covers a hospitalization for an unrelated condition, the compressed representation may introduce noise rather than signal. More sophisticated summarization strategies (e.g., condition-specific summaries) could address this limitation.

P.2 Limitations of the evaluation

Single task. We evaluated on 24-hour ICU mortality prediction. Tasks where distant history is more predictive (chronic disease progression, medication response prediction) may show different patterns.

Single dataset. MIMIC-IV represents a specific patient population (Beth Israel Deaconess Medical Center) and documentation practices. Results may not generalize to other institutions or EHR systems.

Single summarization model. BioClinical-ModernBERT was chosen for its clinical pretraining and long-context support. Other clinical language models (ClinicalBERT, GatorTron) may produce summaries with different characteristics.

Frozen summarizer. We did not fine-tune the summarization model on our task. End-to-end training of the summarizer could improve summary quality but would increase computational costs.

P.3 Future directions

Adaptive summary generation. Current summaries are generated with fixed S values. An adaptive approach could vary summary length based on patient history richness or prediction task requirements.

Multi-task summary reuse. Our preprocessing pipeline generates task-agnostic summaries. Future work could evaluate whether summaries optimized for one task (mortality prediction) transfer to others (readmission, length of stay). If transfer is effective, the amortization benefits would multiply.

Hierarchical summarization. For patients with very long histories (10,000+ events), a single 768-dimensional summary may be insufficient. Hierarchical approaches that summarize at multiple time scales (day, week, hospitalization) could preserve more information while maintaining deployment efficiency.

Summary interpretability. The current summary is a dense vector without interpretable dimensions. Future work could explore structured summaries that preserve identifiable clinical concepts, enabling clinicians to understand what historical information influenced predictions.