LLM Plug-ins Are Not a Free Lunch for Clinical Time-Series Prediction

Juhwan Choi[1*], **Kwanhyung Lee**[1,2*], **Sangchul Hahn**[1], and **Eunho Yang**[1,2]
[1]AITRICS [2]KAIST
{jhchoi, kwanlee9209, steve, eunhoy}@aitrics.com

Abstract

Inspired by recent plug-in frameworks that repurpose frozen layers from large language models (LLMs) as inductive priors, we explore whether such mechanisms can be extended to clinical time-series prediction without textual inputs or LLM fine-tuning. We introduce a lightweight plug-in architecture that inserts a single frozen LLM Transformer layer between an aggregated time-series representation and the prediction head. Unlike prior work focused on vision or language tasks, our study targets clinical time-series data, where LLMs typically underperform when applied directly.

Experiments on two ICU prediction tasks from MIMIC-III show that the proposed plug-in exhibits heterogeneous effects across different backbones and tasks, with occasional performance improvements and minimal computational overhead. We further compare general-purpose and medical-domain LLM layers under an identical plug-in setting, analyzing how domain specialization interacts with clinical time-series models. Overall, our results highlight important limitations of frozen LLM plug-ins and motivate future work on understanding the conditions under which such layers may be beneficial.

1 Introduction

Clinical prediction models play a critical role in supporting clinician decision-making (Steyerberg, 2009). Recent advances in deep learning have substantially improved clinical time-series modeling by enabling architectures capable of capturing complex temporal dependencies and nonlinear interactions in high-dimensional patient data (Xiao et al., 2018; Sun et al., 2020; Morid et al., 2023). However, in contrast to many other domains, large language models (LLMs) have not resulted in consistent gains for clinical forecasting. Across multiple evaluations, LLM-based approaches have often been shown to underperform compared with established, task-specific models (Tan et al., 2024; Chen et al., 2024; Brown et al., 2025). This gap is commonly attributed to the mismatch between the autoregressive, text-oriented design of LLMs and the structured, multivariate, and frequently irregular nature of clinical time-series data (Zhang et al., 2024). In addition, the generative objectives used to train LLMs may not align well with predictive tasks such as clinical risk modeling (Yu et al., 2023).

Despite these limitations, modern LLMs encode extensive medical knowledge that may still be relevant for clinical prediction when incorporated appropriately. State-of-the-art LLMs achieve strong performance across medical question answering benchmarks, including MedQA (Jin et al., 2021), PubMedQA (Jin et al., 2019), and MedMCQA (Pal et al., 2022), in some cases approaching clinician-level accuracy (Labrak et al., 2024; Research and DeepMind, 2025). These results suggest that LLMs internalize clinically meaningful abstractions and domain knowledge that are difficult to acquire solely from structured EHR data (Yan et al., 2025; Hegselmann et al., 2025). However, whether such knowledge can be effectively exploited by small, task-specific clinical models remains largely unexplored. This raises an important open question: rather than replacing specialized clinical time-series models, can LLM-derived representations serve as lightweight inductive priors? If so, how might they influence downstream clinical prediction?

In this work, we explore a simple mechanism for examining the role of LLM-derived priors in clinical time-series prediction. Inspired by recent plug-in frameworks that repurpose single frozen LLM layers as inductive biases in downstream networks (Pang et al., 2024; Kim et al., 2025), we extend this idea to structured clinical prediction tasks.

[*]Equal contribution as co-first author.

Proceedings of the 1st Workshop on Linguistic Analysis for Health (HeaLing 2026), pages 203–211

Our approach inserts a single frozen Transformer layer from an LLM between a backbone model's aggregated time-series representation and its prediction head, using lightweight projection layers to bridge dimensional mismatches. By operating on a single aggregated representation, the proposed plug-in avoids textual inputs, LLM fine-tuning, and sequence-level LLM computation, while remaining compatible with a wide range of clinical time-series architectures.

Through experiments on ICU prediction tasks, we analyze how inserting frozen LLM layers affects predictive behavior across different backbones and tasks. Our experimental results show that inserting frozen LLM layers does not lead to consistent performance gains across tasks and backbones. We therefore analyze potential factors underlying this variability, and use these observations to outline directions for future work on when and how LLM-derived priors may be beneficial for clinical time-series prediction.

Our contributions are summarized as follows:

- We introduce a lightweight plug-in architecture that enables the integration of frozen LLM layers into clinical time-series models without requiring LLM fine-tuning or text-based inputs.
- We present an empirical analysis showing that frozen LLM plug-in layers do not yield consistent performance gains across tasks and backbones, and discuss potential factors underlying this variability to motivate future research directions.

2 Methodology

2.1 Problem Definition

We consider a clinical time-series prediction setting where each patient record is represented as

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}, \quad \mathbf{x}_t \in \mathbb{R}^d,$$

with $\mathbf{x}_t$ denoting d structured clinical variables (e.g., vitals, labs, interventions) observed at time t. Given the observed history $\mathbf{X}_{1:T}$, the goal is to predict a clinical outcome y (e.g., mortality, decompensation, event onset) by estimating

$$p_\theta(y \mid \mathbf{X}_{1:T}),$$

where θ are the parameters of a backbone clinical time-series encoder (e.g., LSTM, Transformer, or any dedicated architecture).

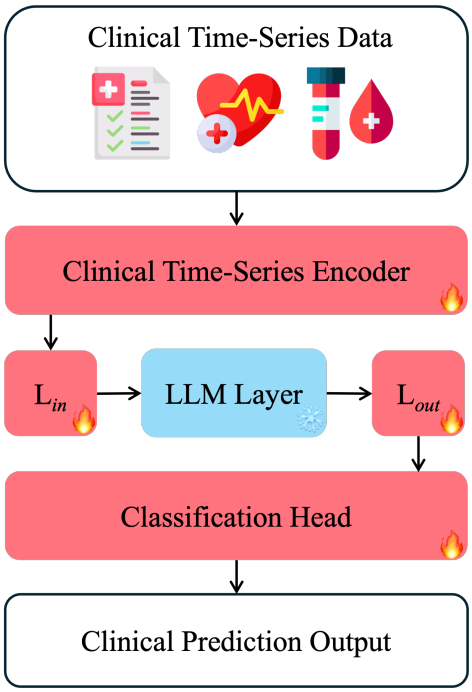

Figure 1: An Illustration of our proposed clinical prediction architecture.

Let Enc_{TS} denote the backbone encoder, Agg an aggregation operator over time (e.g., final state, mean pooling, or a CLS token), and Head the prediction head. The standard baseline pipeline is

$$\mathbf{H}_{ts} = \mathrm{Enc}_{TS}(\mathbf{X}_{1:T}), \qquad \mathbf{h}_{agg} = \mathrm{Agg}(\mathbf{H}_{ts}),$$
$$\hat{y}_{base} = \mathrm{Head}(\mathbf{h}_{agg}).$$

where $\mathbf{H}_{ts}$ is the sequence of hidden states and $\mathbf{h}_{agg} \in \mathbb{R}^{d_{ts}}$ is the aggregated representation. Our objective is to examine the effect of inserting a single frozen LLM layer after the aggregation stage, while keeping the overall framework lightweight.

2.2 Plug-in LLM Layer into Clinical Prediction Models

Following plug-in designs proposed in prior work, we augment the backbone by inserting a single Transformer layer from an LLM (Pang et al., 2024; Kim et al., 2025). Figure 1 illustrates our approach.

Baseline vs. plug-in pipeline. In the baseline, the prediction head directly consumes $\mathbf{h}_{agg}$:

$$\hat{y}_{base} = \mathrm{Head}(\mathbf{h}_{agg}).$$

In our plug-in architecture, we first pass $\mathbf{h}_{agg}$ through a frozen LLM layer, then feed the resulting transformed feature into the same type of prediction head.

Passing the aggregated feature into the LLM. Let L_{LLM} denote a single Transformer layer taken from an LLM with hidden size d_{LLM}. Since $\mathbf{h}_{agg} \in \mathbb{R}^{d_{ts}}$ generally does not match d_{LLM}, i.e., $d_{ts} \neq d_{LLM}$, we introduce two small projection layers:

$$\mathrm{L}_{in} : \mathbb{R}^{d_{ts}} \to \mathbb{R}^{d_{LLM}}, \qquad \mathrm{L}_{out} : \mathbb{R}^{d_{LLM}} \to \mathbb{R}^{d_{ts}}.$$

We first map the aggregated feature into the LLM hidden space,

$$\mathbf{z}_{in} = \mathrm{L}_{in}(\mathbf{h}_{agg}),$$

treat $\mathbf{z}_{in}$ as a single-token input to the frozen LLM layer,

$$\mathbf{z}_{LLM} = \mathrm{L}_{LLM}(\mathbf{z}_{in}),$$

and project it back to the original backbone dimension:

$$\tilde{\mathbf{h}}_{agg} = \mathrm{L}_{out}(\mathbf{z}_{LLM}).$$

The final prediction is then computed as

$$\hat{y} = \mathrm{Head}(\tilde{\mathbf{h}}_{agg}).$$

This design allows the LLM layer to operate only on a single aggregated feature vector, rather than the full sequence, reducing computational overhead while enabling analysis of how LLM-derived transformations interact with clinical time-series representations. Please refer to Appendix A for a pseudo-code of the proposed approach.

Training. During training, we optimize the parameters of Enc_{TS}, Agg (if learnable), L_{in}, L_{out}, and Head using a standard loss (e.g., cross-entropy for classification). All parameters inside L_{LLM} are kept strictly frozen. Consequently, the backbone remains the primary time-series encoder, and the single plug-in LLM layer serves as a lightweight adapter that enriches the clinical representation with LLM-derived clinical knowledge.

3 Experiment

3.1 Experimental Setup

Dataset and Tasks. We evaluate our method on the MIMIC-III database (Johnson et al., 2016) using the benchmark cohort and preprocessing pipeline of Harutyunyan et al. (2019). Specifically, we use the structured time-series benchmarks constructed from ICU stays, where vital signs and laboratory measurements are resampled into hourly intervals and standardized following the released code[1] . Based on this, we adopt the two prediction tasks from the benchmark: in-hospital mortality prediction and decompensation prediction. We use the official train/test split, further split the training set into patient-level train/validation subsets (80:20) for model selection, and select the model checkpoint with the best validation performance for final evaluation on the test set. For evaluation, we follow the standard metrics used in the benchmark: AUROC and AUPRC for both tasks. Appendix D provides further implementation details.

Backbone baselines. We consider three backbone models on the benchmarks. The first model is a Transformer equipped with initial triplet embedding (ITE), following Tipirneni and Reddy (2022). The second model is a standard Transformer applied to hourly resampled time series with last observation carried forward (LOCF) imputation as in prior work on clinical time-series Transformers (Yèche et al., 2021; Lee et al., 2023). Lastly, we adopt an LSTM that uses the same LOCF-based input representation and preprocessing as the second model, providing a recurrent counterpart to the Transformer-based backbone. When we add our plug-in LLM layer, we keep the backbone architecture and optimization hyperparameters fixed to ensure fair comparison.

Research questions. We design our experiments to answer two main research questions:

- **RQ1 (Effect of plug-in LLM layer):** How does inserting a single frozen LLM layer between the aggregated time-series representation and the prediction head affect clinical time-series prediction across different backbones and tasks?
- **RQ2 (Standard vs. medical LLMs):** How do frozen layers from medical-domain LLMs differ from those of general-purpose LLMs under an identical plug-in architecture in clinical time-series prediction?

LLMs used. We consider two different LLM families, each consisting of a medical-domain model and its standard counterpart: (1) **Gemma 3** family, for which we use the original instruction-tuned *Gemma-3-4B-IT* and *MedGemma-4B-IT*, a Gemma 3 variant further adapted for medical data

[1] https://github.com/YerevaNN/mimic3-benchmarks

	In-hospital Mortality		Decompensation	
	AUROC	AUPRC	AUROC	AUPRC
ITE Transformer	0.7865	0.3352	0.8001	0.1418
+ Gemma-3	0.7927	0.3404	0.7865	0.1246
+ MedGemma	0.7899	0.3413	0.7837	0.1195
+ Mistral-7B	0.7898	0.3307	0.7880	0.1372
+ BioMistral-7B	0.7843	0.3350	0.7849	0.1342
LOCF Transformer	0.8204	0.3887	0.8615	0.2478
+ Gemma-3	0.7875	0.3529	0.8810	0.3017
+ MedGemma	0.7914	0.3656	0.8817	0.2774
+ Mistral-7B	0.8197	0.3992	0.8744	0.3129
+ BioMistral-7B	0.8204	0.4003	0.8734	0.3013
LOCF LSTM	0.8428	0.4691	0.8665	0.2601
+ Gemma-3	0.8365	0.4591	0.8835	0.2687
+ MedGemma	0.8365	0.4589	0.8855	0.2719
+ Mistral-7B	0.8359	0.4525	0.8864	0.2855
+ BioMistral-7B	0.8366	0.4655	0.8853	0.2750

Table 1: Experimental results of our proposed method on two clinical prediction tasks. Each experiment has been repeated three times, and the average AUROC and AUPRC are reported. Results that reported a lower mean value than the baseline are gray.

comprehension (Team, 2025; Research and DeepMind, 2025) and (2) **Mistral** family, where we pair the general-purpose *Mistral-7B-Instruct-v0.1* with its biomedical extension *BioMistral-7B*, obtained by further pretraining on large-scale medical corpora (Team, 2023; Labrak et al., 2024). For each family, we extract a layer from both the standard and medical models and plug it into the backbone via the projections described in Section 2.2. Specifically, we use the final layer following previous work (Kim et al., 2025). All LLM parameters are kept strictly frozen; only the backbone encoder, projection layers, and prediction head are trained.

3.2 Result

Table 1 summarizes the results on in-hospital mortality and decompensation prediction, showing that the proposed frozen LLM plug-in does not yield consistent performance improvements, but instead exhibits backbone- and task-dependent effects.

Effect of the plug-in LLM layer (RQ1). Across backbone architectures, the effect of inserting a frozen LLM layer is heterogeneous. For the ITE Transformer, performance changes are generally small and unstable, with occasional gains but frequent degradations, particularly for decompensation prediction. This suggests that when the backbone already encodes strong inductive biases for irregular clinical time series, the additional transformation introduced by a frozen LLM layer may fail to reliably align with task-relevant representations.

In contrast, LOCF-based Transformers and LSTMs show more noticeable—though still non-uniform—effects. Several plug-in configurations improve AUROC and AUPRC for decompensation prediction, especially with the LSTM backbone, while improvements for in-hospital mortality remain limited and inconsistent.

Standard vs. medical-domain LLMs (RQ2). Under identical plug-in settings, medical-domain LLMs do not consistently outperform their general-purpose counterparts. Although medical LLMs occasionally yield better results, particularly on decompensation, the differences are modest and not systematic, indicating that domain specialization alone is insufficient in a frozen, single-layer plug-in setup.

Summary of findings. Overall, frozen LLM layers exhibit inconsistent effects on aggregated clinical representations rather than yielding consistent performance improvements. The observed variability likely reflects systematic factors, including limited trainable capacity in clinical backbones, differences in data scale across tasks, and sensitivity to preprocessing and architectural choices. We further analyze these factors in additional experiments reported in Appendix E.

4 Conclusion

We studied whether frozen LLM layers can serve as lightweight inductive priors for clinical time-series prediction by inserting a single LLM Transformer layer between an aggregated time-series representation and the prediction head. Across two ICU prediction tasks and multiple backbone architectures, we find that frozen LLM plug-in layers do not provide consistent performance gains, but instead interact with clinical time-series representations in a task- and backbone-dependent manner.

Our results suggest that, unlike in vision or language tasks, simply injecting frozen LLM layers into clinical time-series models is insufficient to guarantee improvements. Instead, the effectiveness of LLM-derived priors appears to depend on factors such as model capacity, data scale, and representation compatibility. By highlighting these limitations and failure modes, this work provides empirical guidance on when frozen LLM layers may or may not be beneficial for structured clinical prediction, and motivates future work on more adaptive or data-aware integration strategies.

Limitations

This study has several limitations that should be considered when interpreting the results. First, our plug-in design connects the frozen LLM to the backbone through a single aggregated representation. While this choice ensures computational efficiency, it prevents the LLM from directly interacting with temporal dynamics or multi-step clinical trajectories, which may limit its ability to contribute task-relevant information.

Second, the clinical backbone models considered in this work have relatively limited trainable capacity compared to backbones commonly used in vision or language plug-in settings (Pang et al., 2024; Kim et al., 2025). As a result, the models may lack sufficient flexibility to adapt frozen LLM transformations, particularly in data-scarce tasks such as in-hospital mortality prediction.

Third, our evaluation is restricted to two ICU prediction tasks from the MIMIC-III benchmark. Although widely used, this setting does not capture the full diversity of clinical data distributions, prediction horizons, or real-world deployment conditions. Consequently, the observed effects of frozen LLM layers may not generalize to other clinical tasks or datasets.

Finally, our analysis focuses on frozen, single-layer LLM plug-ins. Rather than aiming to demonstrate universal improvements from frozen LLM layers, our goal is to empirically characterize the conditions under which such plug-in mechanisms are ineffective or fail to provide benefits. We believe that identifying these limitations is a necessary step toward developing more reliable and principled integration strategies for LLMs in clinical time-series modeling.

Ethical Consideration

Integrating a frozen LLM layer into clinical time-series models raises important ethical concerns, particularly regarding bias transfer. LLMs are trained on large-scale text corpora that may encode social biases related to sex, race, age, or socioeconomic status. When a frozen LLM layer is used as a plug-in, such biases cannot be corrected through downstream training and may be implicitly transferred to EHR prediction models via learned projection layers, even without explicit demographic inputs. This may lead to disparate performance or miscalibration across patient subgroups, potentially exacerbating healthcare inequities. We therefore emphasize the need for subgroup-level evaluation, careful model auditing, and transparent reporting when applying frozen LLM components in real-world clinical prediction settings.

References

Katherine E Brown, Chao Yan, Zhuohang Li, Xinmeng Zhang, Benjamin X Collins, You Chen, Ellen Wright Clayton, Murat Kantarcioglu, Yevgeniy Vorobeychik, and Bradley A Malin. 2025. Large language models are less effective at clinical prediction tasks than locally trained machine learning models. *Journal of the American Medical Informatics Association*, 32(5):811–822.

Canyu Chen, Jian Yu, Shan Chen, Che Liu, Zhongwei Wan, Danielle Bitterman, Fei Wang, and Kai Shu. 2024. Clinicalbench: Can llms beat traditional ml models in clinical prediction? In *Proceedings of NeurIPS 2024 The Second Workshop on GenAI for Health Potential, Trust, and Policy Compliance*.

Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. 2018. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *Proceedings of NeurIPS*.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):96.

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

Chao-Chun Hsu, Shantanu Karnwal, Sendhil Mullainathan, Ziad Obermeyer, and Chenhao Tan. 2020. Characterizing the value of information in medical notes. In *Findings of EMNLP*, pages 2062–2072.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of EMNLP*, pages 2567–2577.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Kyeonghyun Kim, Jinhee Jang, Juhwan Choi, Yoonji Lee, Kyohoon Jin, and YoungBin Kim. 2025. Plug-in and fine-tuning: Bridging the gap between small

language models and large language models. In *Proceedings of ACL*, pages 5434–5452.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv Preprint*.

Kwanhyung Lee, John Won, Heejung Hyun, Sangchul Hahn, Edward Choi, and Joohyung Lee. 2023. Self-supervised predictive coding with multimodal fusion for patient deterioration prediction in fine-grained time resolution. In *Proceedings of ICLR 2023 Workshop on Trustworthy Machine Learning for Healthcare*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. 2023. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of CHIL*, pages 248–260.

Ziqi Pang, Ziyang Xie, Yunze Man, and Yu-Xiong Wang. 2024. Frozen transformers in language models are effective visual encoder layers. In *Proceedings of ICLR*.

Google Research and Google DeepMind. 2025. Medgemma technical report. *arXiv Preprint*.

Satya Narayan Shukla and Benjamin M Marlin. 2021. Multi-time attention networks for irregularly sampled time series. In *Proceedings of ICLR*.

Ewout W. Steyerberg. 2009. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*, 1 edition. Springer.

Chenxi Sun, Shenda Hong, Moxian Song, and Hongyan Li. 2020. A review of deep learning methods for irregularly sampled medical time series data. *arXiv Preprint*.

Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. 2024. Are language models actually useful for time series forecasting? In *Proceedings of NeurIPS*, pages 60162–60191.

Gemma Team. 2025. Gemma 3 technical report. *arXiv Preprint*.

Mistral Team. 2023. Mistral 7b. *arXiv Preprint*.

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

Cao Xiao, Edward Choi, and Jimeng Sun. 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428.

Jiahuan Yan, Jintai Chen, Chaowen Hu, Bo Zheng, Yaojun Hu, Jimeng Sun, and Jian Wu. 2025. Small models are llm knowledge triggers for medical tabular prediction. In *Proceedings of ICLR*.

Hugo Yèche, Gideon Dresdner, Francesco Locatello, Matthias Hüser, and Gunnar Rätsch. 2021. Neighborhood contrastive learning applied to online patient monitoring. In *Proceedings of ICML*, pages 11964–11974. PMLR.

Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. 2023. Open, closed, or small language models for text classification? *arXiv Preprint*.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. 2024. Large language models for time series: a survey. In *Proceedings of IJCAI*, pages 8335–8343.

A Pseudo-code of Proposed Method

Algorithm 1 Forward Pass of the Plug-in LLM Layer

Require: Time-series $\mathbf{X}_{1:T}$; backbone encoder Enc_{TS}; aggregation operator Agg

Require: Projection layers $\mathrm{L}_{in}, \mathrm{L}_{out}$; frozen LLM layer L_{LLM}; prediction head Head

$\mathbf{H}_{ts} \leftarrow \mathrm{Enc}_{TS}(\mathbf{X}_{1:T})$ ▷ Encode clinical time series

$\mathbf{h}_{agg} \leftarrow \mathrm{Agg}(\mathbf{H}_{ts})$ ▷ Aggregate temporal states

$\mathbf{z}_{in} \leftarrow \mathrm{L}_{in}(\mathbf{h}_{agg})$ ▷ Project to LLM hidden space

$\mathbf{z}_{LLM} \leftarrow \mathrm{L}_{LLM}(\mathbf{z}_{in})$ ▷ Frozen LLM layer (without gradient flow)

$\tilde{\mathbf{h}}_{agg} \leftarrow \mathrm{L}_{out}(\mathbf{z}_{LLM})$ ▷ Project back to backbone dimension

$\hat{y} \leftarrow \mathrm{Head}(\tilde{\mathbf{h}}_{agg})$ ▷ Final prediction

return $\hat{y}$

B Main Experiment Results with Standard Deviation

Table 2 complements Table 1 with standard deviations of each model.

C Dataset Statistics

We present the dataset statistics of MIMIC-3, preprocessed through Harutyunyan et al. (2019) in Table 3.

	In-hospital Mortality		Decompensation	
	AUROC	AUPRC	AUROC	AUPRC
ITE Transformer	0.7865 ± 0.0022	0.3352 ± 0.0080	0.8001 ± 0.0058	0.1418 ± 0.0027
+ Gemma-3	0.7927 ± 0.0015	0.3404 ± 0.0026	0.7865 ± 0.0046	0.1246 ± 0.0044
+ MedGemma	0.7899 ± 0.0088	0.3413 ± 0.0035	0.7837 ± 0.0040	0.1195 ± 0.0015
+ Mistral-7B	0.7898 ± 0.0050	0.3307 ± 0.0198	0.7880 ± 0.0006	0.1372 ± 0.0058
+ BioMistral-7B	0.7843 ± 0.0016	0.3350 ± 0.0065	0.7849 ± 0.0021	0.1342 ± 0.0043
LOCF Transformer	0.8204 ± 0.0018	0.3887 ± 0.0159	0.8615 ± 0.0119	0.2478 ± 0.0579
+ Gemma-3	0.7875 ± 0.0096	0.3529 ± 0.0156	0.8810 ± 0.0044	0.3017 ± 0.0156
+ MedGemma	0.7914 ± 0.0055	0.3656 ± 0.0099	0.8817 ± 0.0008	0.2774 ± 0.0100
+ Mistral-7B	0.8197 ± 0.0025	0.3992 ± 0.0110	0.8744 ± 0.0016	0.3129 ± 0.0017
+ BioMistral-7B	0.8204 ± 0.0023	0.4003 ± 0.0131	0.8734 ± 0.0050	0.3013 ± 0.0051
LOCF LSTM	0.8428 ± 0.0013	0.4691 ± 0.0074	0.8665 ± 0.0092	0.2601 ± 0.0043
+ Gemma-3	0.8365 ± 0.0022	0.4591 ± 0.0082	0.8835 ± 0.0007	0.2687 ± 0.0070
+ MedGemma	0.8365 ± 0.0007	0.4589 ± 0.0006	0.8855 ± 0.0010	0.2719 ± 0.0052
+ Mistral-7B	0.8359 ± 0.0026	0.4525 ± 0.0130	0.8864 ± 0.0004	0.2855 ± 0.0042
+ BioMistral-7B	0.8366 ± 0.0042	0.4655 ± 0.0124	0.8853 ± 0.0023	0.2750 ± 0.2750

Table 2: Experimental results of our proposed method on two clinical prediction tasks. Each experiment has been repeated three times, and the average AUROC and AUPRC are reported. Results that reported a lower mean value than the baseline are gray.

		Train	Test
In-hospital Mortality	Positive	2,423	374
	Negative	15,480	2,862
Decompensation	Positive	61,013	9,683
	Negative	2,847,401	513,525

Table 3: Dataset statistics of Harutyunyan et al. (2019).

D Implementation Details

Model Architectures. We consider both recurrent- and attention-based backbones. The LSTM backbone consists of two stacked LSTM layers, while the Transformer-based backbone uses four Transformer encoder layers. For all Transformer variants, the dropout rate is fixed to 0.1. Positional encoding is applied only in the Imputation Transformer, following its original design, whereas the ITE Transformer does not employ any explicit positional encoding.

Across all backbones and tasks, the hidden dimension is fixed to 128 to ensure a fair comparison between architectures. This design choice allows us to isolate the effect of architectural differences and the proposed plug-in components, rather than variations in model capacity. All models share a unified MLP-based classifier at the output layer. Specifically, we adopt the classifier architecture proposed in Shukla and Marlin (2021).

All continuous input features are normalized using a feature-wise z-transformation. The mean and standard deviation are computed exclusively on the training data and then applied to the validation and test sets to avoid information leakage.

For time encoding under the ITE setting, we adopt task-specific conventions. In the In-hospital Mortality task, absolute time from ICU admission is used as the temporal input, reflecting the fixed prediction horizon. In contrast, for the Decompensation task, relative time since the last observation is employed to better capture short-term physiological dynamics.

Training and Evaluation Scheme. Unless otherwise stated, all experiments follow a unified training protocol across tasks and architectures. To address severe class imbalance in clinical outcomes, we apply random oversampling of the minority class to achieve an approximately 1:1 class ratio within each training split. Validation and test sets remain untouched and preserve the original class distributions.

Training schedules differ by task. For the In-hospital Mortality task, models are trained for 30

epochs using a linear warmup followed by cosine decay, with the warmup phase lasting for the first 6 epochs. For the Decompensation task, models are trained for 5 epochs with a linear warmup over the first epoch, followed by standard decay. Early stopping is not applied in order to ensure consistent optimization across configurations.

Models are optimized using the AdamW optimizer (Loshchilov and Hutter, 2019), with the learning rate selected via hyperparameter sweep within $[1e{-}3, 1e{-}5]$ and batch size of 128. All reported results are averaged over three random seeds to account for training variability.

E Additional Experiments for Explaining the Inconsistent Gains

In Section 3.2, we observed that inserting a single frozen LLM layer does not lead to consistent performance improvements across tasks, backbones, or preprocessing strategies. Instead, the effect of the plug-in is highly heterogeneous and, in some cases, even detrimental. To better understand the underlying causes of this variability, we design additional experiments targeting two plausible factors: the limited trainable capacity of clinical backbones and the scale of available training data.

E.1 Effect of Trainable Capacity

One potential explanation for the limited and inconsistent gains of the plug-in is the relatively small size of the clinical encoders used in this study. In contrast to prior work in CV and NLP, where frozen pre-trained models are typically combined with large downstream backbones (e.g., ViT or BERT with hundreds of millions of parameters), the clinical backbones employed here contain fewer than one million trainable parameters. Such limited capacity may restrict the model's ability to adapt its representations to effectively utilize a frozen LLM layer.

To test this hypothesis, we conduct an additional experiment on the in-hospital mortality task in which the inserted LLM Transformer layer is trained end-to-end rather than kept frozen. All other architectural components, including the clinical backbone, temporal aggregation module, and prediction head, remain identical to the standard plug-in configuration used in the main experiments. The LLM layer is initialized from the same pre-trained weights as in the frozen setting.

Table 4 demonstrates the results. The results

	In-hospital Mortality	
	AUROC	AUPRC
ITE Transformer	0.7865	0.3352
+ Gemma-3 (Frozen)	0.7927	0.3404
+ Gemma-3 (Trainable)	0.7924	0.3359
+ Mistral-7B (Frozen)	0.7898	0.3307
+ Mistral-7B (Trainable)	0.7911	0.3442
LOCF Transformer	0.8204	0.3887
+ Gemma-3 (Frozen)	0.7875	0.3529
+ Gemma-3 (Trainable)	0.8057	0.3917
+ Mistral-7B (Frozen)	0.8197	0.3992
+ Mistral-7B (Trainable)	0.8170	0.4053
LOCF LSTM	0.8428	0.4691
+ Gemma-3 (Frozen)	0.8365	0.4591
+ Gemma-3 (Trainable)	0.8380	0.4670
+ Mistral-7B (Frozen)	0.8373	0.4529
+ Mistral-7B (Trainable)	0.8375	0.4708

Table 4: Experimental results of our analysis on trainable model parameters. Each experiment has been repeated three times, and the average AUROC and AUPRC are reported. Results that reported a lower mean value than the baseline are gray.

suggest that the trainable plug-in configuration generally yields higher AUPRC than its frozen counterpart across backbones. This trend is particularly noteworthy given that AUPRC is often considered a more informative metric than AUROC in clinical prediction settings with severe class imbalance, as it better reflects performance on rare but clinically critical positive cases (Choi et al., 2018; Hsu et al., 2020). These results suggest that increasing the number of trainable parameters—thereby enhancing representation capacity—enables the model to better adapt and exploit the transformations introduced by the LLM layer, leading to more effective predictive behavior in imbalanced clinical tasks. While the absolute gains remain modest, the consistent improvement in AUPRC indicates that representation flexibility plays a meaningful role in determining the utility of LLM-based plug-ins. Taken together, these findings imply that the limited capacity of the current clinical encoders may be a key bottleneck, and that scaling up the encoder architecture could be a promising direction for enabling more effective integration of LLM-derived representations in future work.

	Decompensation 100%		Decompensation 1%	
	AUROC	AUPRC	AUROC	AUPRC
ITE Transformer	0.7928	0.1498	0.6180	0.0281
+ Gemma-3	0.7853	0.1298	0.6292	0.0306
+ MedGemma	0.7837	0.1195	0.6271	0.0287
+ Mistral-7B	0.7880	0.1372	0.6335	0.0319
+ BioMistral-7B	0.7849	0.1342	0.6384	0.0369
LOCF Transformer	0.8417	0.2304	0.7206	0.0546
+ Gemma-3	0.8570	0.2152	0.7365	0.0666
+ MedGemma	0.8643	0.2189	0.7328	0.0615
+ Mistral-7B	0.8626	0.2528	0.7355	0.0684
+ BioMistral-7B	0.8555	0.2275	0.7246	0.0607
LOCF LSTM	0.8665	0.2601	0.7509	0.0798
+ Gemma-3	0.8835	0.2687	0.7554	0.0807
+ MedGemma	0.8855	0.2719	0.7529	0.0840
+ Mistral-7B	0.8865	0.2855	0.7594	0.0948
+ BioMistral-7B	0.8853	0.2750	0.7634	0.0936

Table 5: Experimental results of our analysis on data scale. Each experiment has been repeated three times, and the average AUROC and AUPRC are reported. Results that reported a lower mean value than the baseline are gray.

E.2 Effect of Data Scale

Another plausible factor contributing to the heterogeneous results is the scale of available training data. As noted in Appendix C, the Decompensation task provides orders of magnitude more training examples than the in-hospital mortality task, raising the possibility that the LLM plug-in requires large-scale supervision to align its representations with task-specific objectives.

To investigate this hypothesis, we perform a controlled data-scaling experiment on the Decompensation task by artificially reducing the size of the training set. Specifically, we construct subsampled training sets using only 1% of the original training data. The test set and evaluation protocol are kept identical to those used in the main experiments, and the same backbone architecture and preprocessing strategy are employed. The LLM plug-in remains frozen, and model initialization follows the standard setup.

As shown in Table 5, reducing the Decompensation training data to 1% reveals several notable patterns that are not apparent in the full-data setting. First, even for the ITE Transformer—where the plug-in failed to improve performance under the full training data—the frozen LLM layer consistently improves both AUROC and AUPRC in the 1% regime. This suggests that the lack of gains observed for ITE under full supervision does not necessarily indicate that the plug-in is ineffective, but rather that ITE may be less suited to exploiting large-scale supervision. Second, although the 1% Decompensation setting becomes quantitatively comparable to the In-hospital Mortality task in terms of training data scale, the observed plug-in behavior differs substantially between the two tasks. While the plug-in exhibits consistent improvements across all backbones for Decompensation under severe data reduction, such a uniform pattern is not observed for In-hospital Mortality in the main experiments. This discrepancy indicates that the effectiveness of frozen LLM plug-ins cannot be explained by data scale alone, and instead points to strong task-specific factors, such as prediction horizon, label definition, and the temporal characteristics of deterioration versus mortality. Together, these results suggest that the interaction between LLM-derived priors and clinical time-series models is jointly governed by backbone inductive bias and task-specific structure, rather than by training data size in isolation.