NEST: NESTED EVENT STREAM TRANSFORMER FOR SEQUENCES OF MULTISETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Event stream data often exhibit hierarchical structure in which multiple events co-occur, resulting in a sequence of multisets (i.e., bags of events). In electronic health records (EHRs), for example, medical events naturally form a sequence of clinical encounters with well-defined temporal structure, while the importance of within-encounter ordering varies across settings: it is clinically meaningful in inpatient and ICU encounters, but potentially trivial in long outpatient trajectories. Most existing foundation models (FMs) for event stream data flatten this hierarchy into a one-dimensional sequence, leading to (i) computational inefficiency associated with dense attention and learning spurious within-set relationships, and (ii) lower-quality set-level representations from heuristic post-training pooling for downstream tasks. Here, we show that preserving the original hierarchy through interleaved within-set and cross-set encoders provides a useful inductive bias that improves both computational efficiency and representation quality. We then introduce **N**ested **E**vent **S**tream **T**ransformer (**NEST**), a FM for event streams comprised of sequences of multisets. Building on this architecture, we formulate Masked Set Modeling (MSM), a paradigm that promotes improved set-level representation learning. Experiments on real-world datasets show that NEST captures real-world dynamics while improving both pretraining efficiency and downstream performance. Our code will be publicized upon acceptance [anonymous link: https://anonymous.4open.science/r/anonymous-code-submit-444B/]

1 INTRODUCTION

Event stream data often have hierarchical structure in which multiple events co-occur. For example, customer purchase histories are sequences of shopping baskets, each of which is an unordered collection of products. In healthcare, structured electronic health records (EHRs) form temporally ordered sequences of healthcare encounters, each of which may be viewed as a collection of medical events. However, the importance of within-encounter ordering varies across healthcare settings: it is clinically meaningful in inpatient and ICU encounters, but can be trivial in long outpatient trajectories. Rather than committing to either regime, we model the event stream as an ordered sequence of event collections, allowing each collection to be ordered or unordered. We call this data structure a **seq**uence of event multi**set**s (`SeqSet`), where a *multiset* is defined as a *set with multiplicity* (*i.e.*, permitting duplicate elements). As we will show, modeling data as a `SeqSet` offers key advantages, namely promoting efficient computation through structured sparsity, and enabling direct learning of set-level representations without post-hoc pooling.

Transformer-based foundation models (FMs) are commonly used to model event-stream data, but typically flat rather than hierarchical Transformer architectures are used (Li et al., 2020; Rasmy et al., 2021; Wornow et al., 2023; Odgaard et al., 2024; Renc et al., 2024), ignoring the hierarchical structure often present. GPT-style causal modeling (McDermott et al., 2023; Renc et al., 2024) imposes a fixed token order on groups of occurred events, while BERT-style full attention (Rasmy et al., 2021; Odgaard et al., 2024) preserves all pairwise interactions but at computing cost that grows quadratically with context size. More fundamentally, both paradigms learn representations only at the token level through MLM or NTP; whereas set-level representations, which are often more salient for EHRs, shopping carts, and other hierarchical event streams, are only recovered through post-hoc pooling, which prior work has shown to be suboptimal (Reimers & Gurevych, 2019; Gao

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

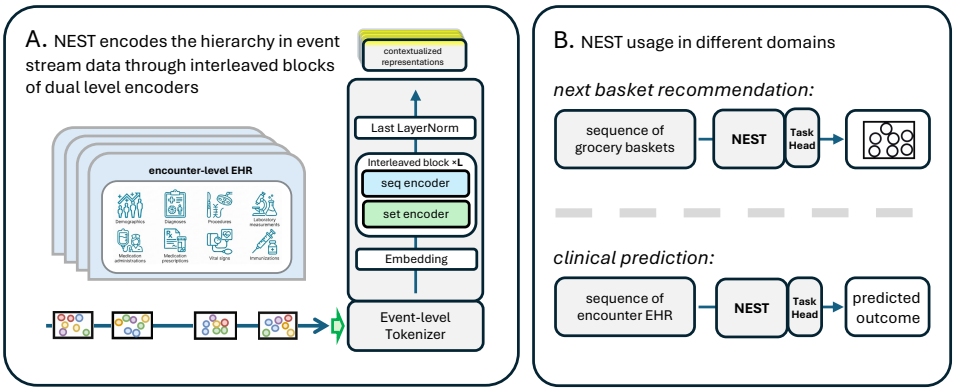

Figure 1: NEST is a FM for event streams structured as sequences of multisets (`SeqSet`). **(A)** Each NEST layer is an interleaved block consisting of a Set-Wise Encoder (SWE) followed by a Cross-Set Encoder (CSE), repeated L times. **(B)** The same architecture instantiates across domains.

et al., 2021). The principle of learning representations of higher-level, semantically coherent units rather than individual tokens has been advocated in language modeling (Barrault et al., 2024) and object-centric visual reasoning (Locatello et al., 2020). We instantiate this idea for hierarchical event streams through joint architectural and pretraining changes, treating multisets as the natural unit of representation, in contrast to existing FMs that encode flat event sequences of EHR events (Li et al., 2020; Wornow et al., 2023), purchase histories (Kang & McAuley, 2018; Sun et al., 2019), and financial transactions (Hu et al., 2023) as atomic tokens.

Building on hierarchical models for text (Yang et al., 2016; Chalkidis et al., 2022) and event streams (Hu & He, 2019; Li et al., 2022), we introduce the **N**ested **E**vent **S**tream **T**ransformer (**NEST**), a foundation model that is structurally aligned with `SeqSet` data. Each NEST layer pairs a set-wise encoder (SWE) for within-set token interactions with a cross-set encoder (CSE) for sequence-level dynamics, with the two encoders interleaved layer-by-layer rather than stacked in two phases. This interleaving yields *dual-level contextualization*, providing (i) structurally induced sparse attention and (ii) direct set-level representations without post-hoc pooling.

To train NEST, we propose a novel pretraining objective, Masked Set Modeling (MSM), that is tailored for sequences of multisets and complements the standard MLM objective during NEST training. The MSM objective enables NEST to learn ready-to-use set representations, eliminating heuristic post-training pooling.

We conduct extensive experiments across event stream datasets from multiple real-world domains to evaluate NEST's effectiveness. NEST achieves state-of-the-art performance and improved computational efficiency compared to Transformer models trained on flattened event streams across diverse downstream tasks. In addition, ablation studies show that both NEST's interleaved architecture and proposed MSM objective enhances set-level representation learning and thus yield better performances in downstream task. We argue that the right inductive bias for event-stream foundation models is to make both the architecture and pretraining objective reflect the data's nested set structure rather than ignoring it by flattening the sequence. Our contributions are:

- **Natural architectural for SeqSet data.** We introduce **NEST**, a hierarchical Transformer foundation model whose interleaved within-set and cross-set encoders respect SeqSet's opposed symmetries: intra-set exchangeability and inter-set temporal order.
- **Architectural rationale for interleaving.** We articulate an information-theoretic perspective on interleaved versus sequential composition and validate its predictions in various experiments. We find compression should not precede cross-set context.
- **Set-level pretraining objective.** We formulate **Masked Set Modeling (MSM)**, a masked reconstruction task that complements MLM: a masked multiset's token distribution is reconstructed *from its surrounding sets within the same stream*, with `[CLS]` serving as the dedicated set-summary token that carries this cross-set information.

2 Related Works

Hierarchical Transformer and Structured Attention. Hierarchical architectures have proven effective for modeling structured text (Chalkidis et al., 2019; 2022; He et al., 2024), achieving efficiency through structurally induced sparsity. Alternative approaches to efficient attention include sliding-window patterns (Beltagy et al., 2020; Zaheer et al., 2020), kernel-based or landmark approximations (Choromanski et al., 2020; Xiong et al., 2021), and learned latent bottlenecks (Jaegle et al., 2021). However, these methods trade off attention fidelity for efficiency without leveraging data's natural grouping structure.

EHR Foundation Models. Early EHR FMs adopted BERT-style MLM (Li et al., 2020; Rasmy et al., 2021), with subsequent work exploring autoregressive (Wornow et al., 2023; Renc et al., 2024), time-to-event (Steinberg et al., 2023), and contrastive (Jeong et al., 2023) objectives. Architecturally, CORE-BEHRT (Odgaard et al., 2024) optimizes the flat Transformer but collapses all multisets into a single sequence. HEART (Huang et al., 2024) enriches within-encounter representations with additional pairwise relational embeddings. Hi-BEHRT (Li et al., 2022) and GT-BEHRT (Poulain & Beheshti, 2024) adopt sequential two-stage designs: Hi-BEHRT uses fixed sliding windows that can fragment multisets, and GT-BEHRT decouples a per-visit graph encoder from a sequence Transformer during pretraining. NEST instead interleaves within-set and cross-set processing, respects multiset boundaries, and introduces MSM for explicit set-level representation learning.

Set-level Representation. Naive pooling strategies such as mean-pooling or directly using `[CLS]` embeddings yield suboptimal set-level representations (Reimers & Gurevych, 2019). Principled alternatives include Pooling by Multihead Attention (Lee et al., 2019) for permutation-invariant set summarization and contrastive methods for refining `[CLS]` embeddings (Gao et al., 2021; Chuang et al., 2022). The latter, however, shape the embedding geometry without ever requiring `[CLS]` to reconstruct the set's contents, so set summarization is supervised only implicitly for the token to summarize set contents. NEST instead explicitly supervises set-level representations through a set prediction objective (MSM, Section 3.5), using the `[CLS]` token as the set-level readout.

3 Method

3.1 Event Stream as a Sequence of Sets

We model an event stream as a **time-ordered sequence of multisets**, where each multiset represents a collection of events that may be time-ordered (e.g., short ICU settings) or treated as unordered (e.g., long-term routine clinical visits). Formally:

$$\big((\mathcal{X}_i, T_i)\big)_{i=1}^{m}, \;\; \mathcal{X}_i = \big\{(\boldsymbol{c}_i, T_i)\big\} \cup \big\{(\mathbf{x}_{ij}, t_{ij})\big\}_{j=1}^{n} \tag{1}$$

where $\mathcal{X}_i$ is the i-th event multiset (e.g., clinical visit), padded and truncated to fixed size n, and m is the number of multisets. Each set includes a learnable `[CLS]` token $\boldsymbol{c}_i$ that serves as the set-level readout. We assign $\boldsymbol{c}_i$ the set timestamp T_i so that the CSE can encode inter-set time ordering via RoPE (Section 3.3). The optional fine-grained timestamps t_{ij} capture within-set temporal structure when available.

We define embedding notation used throughout. Let $[\cdot]$ denote column-wise concatenation. $X_i = [\{\boldsymbol{c}_i\} \cup \{\boldsymbol{x}_{ij}\}_{j=1}^{n}] \in \mathbb{R}^{d_{\text{model}} \times (n+1)}$ represents the token embeddings of $\mathcal{X}_i$. At the l-th NEST layer, $\boldsymbol{c}_i^{(l)}$ and $Z_i^{(l)} = [\{\boldsymbol{z}_{ij}^{(l)}\}]$ denote the contextualized `[CLS]` and non-`[CLS]` hidden states, respectively. We drop the superscript for final-layer outputs: $\boldsymbol{c}_i \coloneqq \boldsymbol{c}_i^{(L)}$, $Z_i \coloneqq Z_i^{(L)}$.

3.2 Overview of NEST

Each NEST layer consists of a Set-Wise Encoder (SWE) followed by a Cross-Set Encoder (CSE), interleaved layer-by-layer. Although prior work has empirically explored such interleaving for long documents (Chalkidis et al., 2022), the architectural consequences for SeqSet data and set-level pretraining have not been characterized. We provide an information-theoretic analysis in Section 3.4 and simulation study in Section 4.2. Within each layer, the two encoders perform complementary roles: tokens attend only within their own multiset via SWE; cross-set information flows exclusively

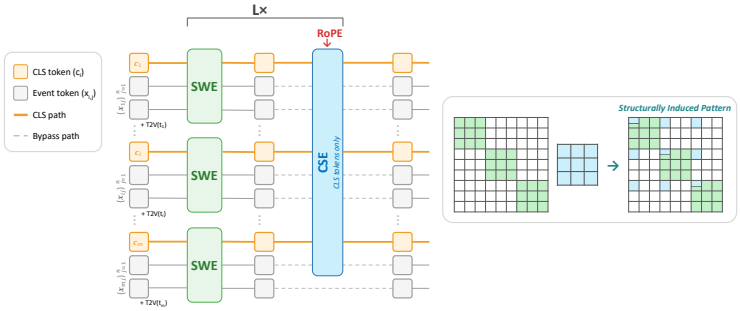

Figure 2: NEST models event stream data as a sequence of event multisets (`SeqSet`). At each NEST layer, a SWE contextualizes tokens within each set, followed by a CSE that contextualizes `[CLS]` tokens across sets with RoPE-encoded inter-set time. The two encoders are interleaved. Non-`[CLS]` token hidden states are propagated through bypass paths so that within-set information remains accessible at every layer. The composition yields a structurally induced sparse attention pattern.

through the `[CLS]` tokens via CSE. Although SWE and CSE each apply dense attention internally, their composition over the full token set induces a sparse attention pattern (Figure 2, right). *Across the full stack, global context emerges through the `[CLS]`-mediated bridge: CSE injects cross-set context into `[CLS]`, and the subsequent SWE distributes it back to set members*. This design (i) improves computing efficiency through structural sparsity, and (ii) produces set-level representations via dual-level contextualization without post-hoc pooling.

3.3 Dual-level Contextualization

We use modern Transformer components (Warner et al., 2025) in all SWE/CSE blocks. Within each multiset, interactions are restricted to in-set tokens, with the `[CLS]` aggregating set-level features via set-wise contextualization. Across the sequence, `[CLS]` communicate through cross-set contextualization.

Set-wise. The set-wise contextualization occurs within each set independently. Given the multiset $\mathcal{X}_i$,

$$\begin{aligned} \boldsymbol{z}_{ij}^{(l+1)} &\leftarrow \boldsymbol{z}_{ij}^{(l)} + \sum_a W_O^{(l,a)} V_i^{(l,a)} \text{Softmax}\left(\frac{1}{\sqrt{d_k}} K_i^{(l,a)\top} q_{ij}^{(l,a)}\right) \\ V_i^{(l,a)} &= W_V^{(l,a)} X_i^{(l)} = [\{\boldsymbol{v}_{ij}^{(l,a)}\}_{j=1}^n] \\ K_i^{(l,a)} &= W_K^{(l,a)} X_i^{(l)} = [\{\boldsymbol{k}_{ij}^{(l,a)}\}_{j=1}^n] \end{aligned} \tag{2}$$

$V_i^{(l,a)}$ and $K_i^{(l,a)}$ are the layer- and head-specific value and key matrices for the tokens in $\mathcal{X}_i$, respectively, and $q_{ij}^{(l,a)} = W_Q^{(l,a)} \boldsymbol{z}_{ij}^{(l)}$ is the layer- and head-specific query vector of token j. $W_O^{(l,a)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_Q^{(l,a)}, W_K^{(l,a)}, W_V^{(l,a)} \in \mathbb{R}^{d_k \times d_{\text{model}}}$ are the head-specific projection matrices in multi-head attention of SWE. SWE omits positional encoding by default, enforcing permutation invariance over intra-set tokens. When fine-grained timestamps are available or necessary (e.g., ICU and inpatient settings), an optional intra-set RoPE can be applied to keys analogously to CSE, encoding within-set temporal order via t_{ij}. Inside each NEST block, SWE is shared across all multisets in the sequence.

Cross-set. The cross-set contextualization operates solely on `[CLS]` embeddings $\{\boldsymbol{c}_i^{(l)}\}_{i=1}^m$. The hidden states of the other tokens bypass CSE.

$$\begin{aligned} \boldsymbol{c}_i^{(l+1)} &\leftarrow \boldsymbol{c}_i^{(l)} + \sum_a \tilde{W}_O^{(l,a)} \tilde{V}^{(l,a)} \text{Softmax}\left(\frac{1}{\sqrt{d_k}} \tilde{K}^{(l,a)\top} R(\texttt{pos}_i) \tilde{q}_i^{(l,a)}\right) \\ \tilde{V}^{(l,a)} &= \tilde{W}_V^{(l,a)} [\{\boldsymbol{c}_i^{(l)}\}_{i=1}^m] = \left[\{\tilde{\boldsymbol{v}}_i^{(l,a)}\}_{i=1}^m\right] \\ \tilde{K}^{(l,a)} &= \text{RoPE}\left(\tilde{W}_K^{(l,a)} [\{\boldsymbol{c}_i^{(l)}\}_{i=1}^m]\right) = \left[\{R(\texttt{pos}_i) \tilde{\boldsymbol{k}}_i^{(l,a)}\}_{i=1}^m\right] \end{aligned} \tag{3}$$

where $\tilde{q}_i^{(l,a)} = \tilde{W}_Q^{(l,a)} \boldsymbol{c}_i^{(l)}$ is the query vector for the [CLS] token in $\mathcal{X}_i$, $\tilde{W}_O^{(l,a)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $\tilde{W}_Q^{(l,a)}, \tilde{W}_K^{(l,a)}, \tilde{W}_V^{(l,a)} \in \mathbb{R}^{d_k \times d_{\text{model}}}$ are the head-specific projection matrices in the multi-head attention of CSE, and $R(\texttt{pos}_i)$ is the rotary matrix for the position scalar $\texttt{pos}_i$. By default, we set $\texttt{pos}_i = i$ (set index); when inter-set timestamps are available and informative (e.g., calendar time between encounters), $\texttt{pos}_i$ can instantiate as T_i to encode temporal intervals directly.

The [CLS] tokens act as inter-set routers: collecting set-specific features in SWE, communicating across sets in CSE, and dispatching cross-set context back to non-[CLS] tokens in the subsequent SWE. This cycle repeats across layers, with $\{\boldsymbol{c}_i\}_{i=1}^m = \{\boldsymbol{c}_i^{(L)}\}_{i=1}^m$ serving as the canonical multiset representations for downstream tasks.

3.4 NEST Information Bypass Intuition

By the Data Processing Inequality, information discarded at an intermediate stage cannot be recovered downstream. This separates two ways of arranging SWE and CSE blocks. A *sequential* design stacks all SWE blocks first; at the SWE–CSE boundary the set is pooled into a single [CLS] embedding and the token-level state is discarded. NEST instead *interleaves* the two block types, keeping token states on bypass paths throughout. Appx A gives more details for the comparison. We offer this as intuition rather than a formal claim; it is supported by a controlled simulation (Section 4.2) and that the matched sequential variant consistently underperforms NEST across MIMIC-IV tasks (Section 5).

3.5 Dual-level Pretraining Procedure

The interleaved architecture (Section 3.4) provides a channel for set-level information to flow through [CLS], but does not enforce that [CLS] actually summarizes its set's contents. We close this gap with a pretraining objective that explicitly requires [CLS] to predict its multiset's token distribution, complementing the standard MLM objective.

Masked Set Modeling (MSM). Let $\mathcal{V}$ be the token vocabulary. We model each multiset $\mathcal{X}_i$ as a draw from $Mult(n;\, \pi_\theta(\cdot \mid \{\boldsymbol{c}_{i'}\}_{i' \neq i}))$ over $\mathcal{V}$, where $n = |\mathcal{X}_i|$ is the set size and the rate vector π_θ lies in the simplex $\Delta^{|\mathcal{V}|-1}$. This is a modeling simplification: it discards within-set co-occurrence structure but retains the dominant signal of which token types are present at what frequencies, which is what set-level downstream tasks require. The MSM loss maybe viewed as the KL divergence between the empirical and predicted token distributions, with the predicted distribution read out from the final-layer [CLS] embedding $\boldsymbol{c}_i$, which has been contextualized with $\{\boldsymbol{c}_{i'}\}$:

$$\mathcal{L}_{\text{MSM}} = D_{\text{KL}}(\hat{P}_i \,\|\, \pi_\theta), \qquad \pi_\theta = \text{MLP}(\boldsymbol{c}_i), \qquad \text{MLP} : \mathbb{R}^{d_{\text{model}}} \to \Delta^{|\mathcal{V}|}, \tag{4}$$

where $\hat{P}_i$ is the empirical distribution from token-frequency counts in $\mathcal{X}_i$. We provide the full derivation from the multinomial likelihood in Appendix B.

Complementarity with MLM. MLM and MSM operate at distinct granularities and are computed in parallel during pretraining. MLM masks 20% of individual tokens, supervising fine-grained token-level contextualization through SWE. MSM masks entire multisets with probability 40%: within an MSM-masked set, all non-special tokens (including [PAD], excluding [CLS]) are replaced with [MASK]. This full-set masking prevents information leakage from the MLM protocol (Devlin et al., 2019; Odgaard et al., 2024): if any non-[CLS] token were left visible within an MSM-masked set, SWE could reconstruct $\hat{P}_i$ locally without forcing [CLS] to engage cross-set context. The overall training loss is $\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{MSM}}$. Together, these two objectives encourage [CLS] to aggregate set-level information through cross-set context, yielding embeddings $\{\boldsymbol{c}_i\}$ that are usable for set-level downstream tasks without post-hoc pooling. Appendix B.4 ablates this design and shows that each objective works through the path it trains: MLM through the token states and MSM through [CLS].

4 Experiments

In this section, we evaluate NEST through three sets of experiments: an efficiency benchmark against efficient Transformer variants, a controlled simulation comparing the interleaved design to its sequential counterpart, and task performance on diverse real-world datasets.

4.1 NEST Computing Efficiency

We compare NEST's computing efficiency with other Transformer architectures. For fairness, these counterparts are also updated using the ModernBERT building blocks (Warner et al., 2025). Table 1 provides the benchmark results of models' computing efficiency on randomly generated batches of `SeqSet` data. We use the same set of architectural hyperparameters for all tested models: $L = 6, d_{\text{model}} = 768, d_h = 2048, n_{\text{heads}} = 12, d_k = 64, |\mathcal{V}| = 45$K, batch size of 16 and context size of 2048 ($n = 32, m = 64$).

All metrics are normalized to the token level. Despite a larger parameter count, NEST requires fewer FLOPs/token, achieves significantly higher throughput (paired t-test, $p < 0.05$), and consumes less peak memory. NEST's efficiency gains come from attention complexity reduction, not from being a smaller model. Moreover, unlike the approximate attention methods (Choromanski et al., 2020; Xiong et al., 2021) in the comparison, NEST computes *exact* attention, so the efficiency gains reflect architectural effect rather than approximation error tradeoff. NEST uses Memory Efficient Attention (MEA) (Lefaudeux et al., 2022) for both SWE and CSE.

Table 1: Benchmarking Transformer forward-pass efficiency on a single NVIDIA H200 GPU node. N denotes the total context length. Longformer uses window size w and m global anchors; Performer uses random feature dimension size of $r = [d_k \log(d_k)]$; Nyströmformer uses m landmarks; NEST uses m sets with per-set size n.

model	Attn time cmplx	GFLOPs/tok ↓	K tok/s ↑	params (M)	Peak mem (GiB) ↓
BERT	$O(N^2)$	0.1918	$58.89_{\pm 3.04}$	77	52.8
BERT_{MEA} (Lefaudeux et al., 2022)	$O(N^2)$	0.1918	$65.25_{\pm 4.58}$	77	35.9
Longformer (Beltagy et al., 2020)	$O(Nw + Nm)$	0.1714	$69.11_{\pm 5.05}$	88	43.1
Performer (Choromanski et al., 2020)	$O(Nr)$	0.1639	$77.86_{\pm 5.19}$	77	44.5
Nyströmformer (Xiong et al., 2021)	$O(Nm)$	**0.1575**	$88.79_{\pm 7.26}$	77	36.7
NEST	$O(Nn + m^2)$	**0.1573**	**90.46**$_{\pm 6.99}$	120	**34.9**

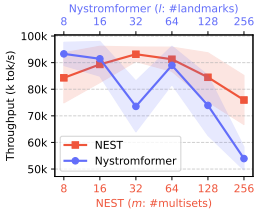

Figure 3: Nyströmformer suffers a sharper computational cliff at larger l.

4.2 Advantages of NEST's Interleaved Design: A Simulation Study

Our intuition predicts that the interleaved design preserves strictly more information than the sequential design. However, it remains to show that this advantage translates to task performance and to characterize when it is most pronounced. Since real-world comparisons confound topology with other design choices, we construct a controlled simulation that isolates topology as the sole variable.

Clinical outcomes often depend on the interaction between individual events and a patient's broader longitudinal context. We design a simulation study to examine interleaved design's advantage. The full Data Generation Process details are in Appendix D.2. We define a synthetic outcome: $y = \sum_{ij} \text{softmax}_j\big(\langle \boldsymbol{x}_{ij}, \boldsymbol{s}_i \rangle\big) \cdot \langle \boldsymbol{x}_{ij}, \boldsymbol{s}_i \rangle$, where $\boldsymbol{s}_i$ is the cross-set context summary for set i and $\boldsymbol{x}_{ij}$ is the token embedding. The outcome measures how much content the sets share, high when the same token repeat within and across sets, near zero when the sets are completely different.

We compare parameter-matched architectures with (a) L SWE layers interleaved with L CSE layers, and (b) L SWE layers followed by L CSE layers, for depth $L \in \{2, 4, 6\}$ and at varying noise scales. Fig. 4 shows that the interleaved design consistently outperforms the sequential design, with the gap widening at greater depth. The sequential design exhibits diminishing or even negative returns from added depth. We observe the same pattern for scenario with uniform codebook (Fig. D1).

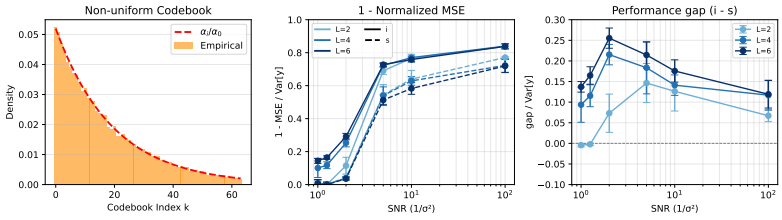

Figure 4: Comparing interleaved design with the ad-hoc sequential design.

4.3 Real-World Data Tasks

We conduct experiments on three real-world datasets: proprietary EHR data from our institution and two publicly available datasets: MIMIC-IV-hosp and Instacart. Table 2 summarizes dataset statistics.

Instacart. This dataset is open sourced by Instacart[1]. The dataset consists of online grocery transactions used to develop model for Next Basket Recommendation (NBR). Orders are treated as baskets containing unordered and equally important items (Li et al., 2023b), so we turn off the within-set RoPE.

Table 2: Real world datasets' statistics.

Dataset	within-set RoPE	#subject	#sets	#tokens
Instacart	off	206K	3.42M	33.8M
MIMIC-IV hosp	on	223K	546K	122M
Proprietary EHR	off	500K	7.34M	70.9M

MIMIC-IV `hosp`. MIMIC-IV is a public dataset of de-identified EHR from Beth Israel Deaconess Medical Center (Johnson et al., 2023). We use version 3.1, which spans admissions from 2008 to 2022. We extract data from the `hosp` module across six source tables: `admissions`, `diagnoses_icd`, `procedures_icd`, `labevents`, `emar`, and `prescriptions`. We adapt the ETHOS's (Renc et al., 2024) tokenization protocol with adjustments for the MIMIC-IV clinical event streams to avoid data leakage. Appendix E.2 has more details.

Proprietary institutional EHR. We developed an ETL pipeline that transforms heterogeneous tables from our institutional EHR database into a unified event-stream format. We collected medical event stream data from approximately 0.5M patients born between 1999 and 2024 and filtered events occurring between ages 0 and 18. Modeling the long-term patient timelines, we ignore the order of events within each encounter. We assign patients born on the 17th–24th of each month to the test set and the remainder to the training and validation splits.

4.3.1 Instacart: Next-Basket Recommendation

Given a user's purchase history $\big((\mathcal{X}_i, T_i)\big)_{i=1}^{m}$, next basket recommendation (NBR) predicts the set of items $\mathcal{X}_{m+1}$ the user will purchase next. Instacart exhibits a high repeated purchase rate (Li et al., 2023b), making repeat-aware methods, such as Sets2Sets (Hu & He, 2019), TIFUKNN (Hu et al., 2020), and ReCANet (Ariannezhad et al., 2022), particularly effective. We use this task to evaluate whether NEST's pretrained set-level representations transfer to downstream applications.

Table 3: Comparison on Instacart NBR.

Model	Recall@10	NDCG@10
Repeat-aware		
Sets2Sets	$28.69_{\pm 0.09}$	$35.00_{\pm 0.38}$
TIFUKNN	34.35	41.19
ReCANet	$34.48_{\pm 0.01}$	$42.60_{\pm 0.03}$
Pretrained FM (zero-shot)		
BERT4Rec	$26.07_{\pm 0.45}$	$31.57_{\pm 0.38}$
NEST	$29.53_{\pm 0.32}$	$35.96_{\pm 0.25}$
FM + Repeat-aware probing		
BERT + ReCANet	$34.70_{\pm 0.04}$	$42.98_{\pm 0.02}$
NEST + ReCANet	$\mathbf{34.82}_{\pm \mathbf{0.03}}$	$\mathbf{43.12}_{\pm \mathbf{0.02}}$

We consider two settings. First, following the BTBR inference protocol (Li et al., 2023a), we append a query basket $\mathcal{X}_{m+1} = \{\texttt{[MASK]}\}$ and rank items by logit at the mask position (no fine-tuning). Second, we replace ReCANet's randomly initialized embeddings with NEST's pretrained representations: NEST's embedding layer serves as the item embedding, and NEST's contextualized `[CLS]` representations serve as the user embedding.

Table 3 shows that NEST alone is competitive with BERT4Rec (Sun et al., 2019; Li et al., 2023a) but underperforms state-of-the-art repeat-aware baselines, as expected given that its pretraining objective does not explicitly model repeat purchases. However, NEST + ReCANet outperforms all baselines on NDCG@10, indicating that NEST's pretrained representations capture richer user-item dynamics than random initialization.

4.3.2 MIMIC-IV Hosp

We compare NEST with baselines covering major EHR FM paradigms: Hi-BEHRT (Li et al., 2022), CEHR-BERT (Pang et al., 2021), MOTOR (Steinberg et al., 2023), CORE-BEHRT (Odgaard et al., 2024), GT-BEHRT (Poulain & Beheshti, 2024), and HEART (Huang et al., 2024), all built with

[1]While the official source is no longer directly hosted, a copy of the dataset is available on Kaggle: *instacart-online-grocery-basket-analysis-dataset*.

matched architecture hyperparameters for fair comparison. The evaluation covers 5 downstream tasks: inpatient mortality prediction, 30-day readmission, prolonged length of stay (PLoS), ICD chapter classification, and ICD category multi-label classification. All models receive only the patient's prior admissions and the target admission truncated at a task-specific prediction time: 24 hours after admission for mortality, 48 hours for PLoS, and discharge for the remaining three tasks. Lab events are timestamped by report time rather than specimen collection, so the input at prediction time mirrors exactly what a clinical decision-support system could observe. Since all tasks except for readmission are within-encounter predictions, we take the patient representation computed from this time-truncated $\mathcal{X}$. Detailed task definitions and experiment protocols are provided in Appendix E.2.1.

Table 4: MIMIC-IV finetuned downstream performance (mean ± SD across 5 random seeds). Avg. rank is the mean rank of each model over the five tasks.

(a) AUROC

Model	Mortality	Readmission	PLoS	ICD Chapter	ICD Category	Avg. rank
Hi-BEHRT	87.48±0.36	70.34±0.22	73.18±0.19	86.09±0.97	84.38±0.40	8.0
MOTOR	88.92±0.57	73.33±0.36	76.17±0.09	89.22±0.82	88.69±0.05	5.6
CEHR-BERT	89.75±0.22	72.36±0.35	76.22±0.17	90.67±0.51	88.57±0.19	5.2
CORE-BEHRT	90.06±0.35	73.47±0.57	77.29±0.08	90.81±0.33	88.97±0.06	3.6
GT-BEHRT	90.30±0.07	72.17±0.29	75.29±0.05	89.16±0.66	88.05±0.07	6.0
HEART	89.51±0.26	71.23±0.42	76.88±0.19	91.86±0.30	89.54±0.14	4.4
$\text{NEST}_{\text{MLM only}}$ (Ours)	91.57±0.20	74.22±0.14	79.38±0.17	92.42±0.22	89.41±0.23	2.2
NEST (Ours)	**92.05**±0.16	**74.75**±0.35	**79.88**±0.08	**92.92**±0.42	**90.20**±0.22	1.0

(b) AP

Model	Mortality	Readmission	PLoS	ICD Chapter	ICD Category	Avg. rank
Hi-BEHRT	18.94±1.31	38.81±0.44	46.63±0.58	37.23±2.39	10.29±0.19	8.0
MOTOR	29.10±0.75	44.80±0.60	60.07±0.29	46.15±0.55	13.16±0.16	4.8
CEHR-BERT	30.34±1.51	43.64±0.86	59.20±0.32	47.99±1.25	13.43±0.11	5.2
CORE-BEHRT	32.60±0.50	44.72±0.55	59.85±0.20	50.10±1.52	13.82±0.20	3.8
GT-BEHRT	28.74±1.26	44.55±0.58	56.99±0.34	48.69±0.39	12.61±0.18	6.0
HEART	27.05±1.27	43.28±0.36	58.84±0.16	52.56±0.66	15.66±0.15	5.2
$\text{NEST}_{\text{MLM only}}$ (Ours)	36.88±1.20	48.06±0.41	62.39±0.19	54.18±0.78	16.57±0.05	2.0
NEST (Ours)	**37.80**±1.09	**48.61**±0.56	**62.81**±0.22	**54.95**±0.83	**16.86**±0.13	1.0

4.3.3 PROPRIETARY EHR

In downstream tasks, we obtain patient representation by encoding the truncated prefix $\big((\mathcal{X}_i, T_i)\big)_{\leq a}$ to predict targeted clinical events occurring after the discharge time T_a. This calculation examines FMs' performances on preventive risk prediction.

We curated three binary classification cohorts: (i) Recurrent Acute Otitis Media (rAOM) (22%), (ii) Autism Spectrum Disorder (ASD) (2.1%), (iii) Home Health Care 30-day readmission (20%). More details of the data pipeline and cohorts' clinical significance are provided in Appendix E.3. These tasks focus on pediatric cohorts, aiming to support in-house clinical decision-making for preventive care. We benchmark our model's performance on these tasks against count-based LightGBM (Ke et al., 2017; Wornow et al., 2023) and CORE-BEHRT (Odgaard et al., 2024), the best performing baseline from MIMIC-IV. In Table 5, for foundation models, we report performance under both nonlinear probing (with the backbone frozen, labeled with `pb`) and full fine-tuning (with the backbone adapted to the downstream task, labeled with `ft`).

Table 5: Performance on proprietary EHR downstream tasks.

Model	rAOM		ASD		30d Readmit	
	AUROC	AP	AUROC	AP	AUROC	AP
LightGBM	70.91±0.71	41.10±0.81	73.36±0.36	6.04±0.24	73.88±0.05	47.00±0.12
$\text{CORE-BEHRT}_{\texttt{pb}}$	71.58±0.39	39.01±0.65	70.03±0.29	9.35±0.46	73.31±0.32	46.74±0.30
$\text{CORE-BEHRT}_{\texttt{ft}}$	73.39±0.39	**42.35**±0.64	73.96±0.76	10.91±0.42	74.07±0.19	47.53±0.33
$\text{NEST}_{\texttt{pb}}$	72.86±0.33	40.50±0.51	71.90±0.59	9.45±0.43	73.31±0.06	46.50±0.24
$\text{NEST}_{\texttt{ft}}$	**73.54**±0.38	41.26±0.65	**74.16**±0.76	**12.05**±0.60	**74.39**±0.15	**47.56**±0.36

5 Ablation Studies

We probe NEST's design on three MIMIC-IV tasks (mortality, 30-day readmission, prolonged length of stay). Full setup is in Appendix G, and Table 6 summarizes the results.

Table 6: Ablation study of NEST on MIMIC-IV. Ablation variants report mean ± SD across 3 random seeds (the two NEST rows repeat the 5-seed results in Table 4). Architectural ablations (Group B) require re-pretraining; behavioral ablations (Group A) reuse the NEST (MLM+MSM) checkpoint and only modify fine-tuning. The "− CSE" and "− SWE" variants are pretrained with MLM, like NEST (MLM only); Sequential is pretrained with MLM and MSM, like NEST (full). Variants marked with † are parameter-matched to the NEST encoder (∼101M).

		Mortality		Readmission		PLoS	
	Variant	AUROC	AP	AUROC	AP	AUROC	AP
	NEST (full)	**92.05**$\pm$**0.16**	**37.80**$\pm$**1.09**	**74.75**$\pm$**0.35**	**48.61**$\pm$**0.56**	**79.88**$\pm$**0.08**	**62.81**$\pm$**0.22**
	NEST (MLM only)	91.57$\pm$0.20	36.88$\pm$1.20	74.22$\pm$0.14	48.06$\pm$0.41	79.38$\pm$0.17	62.39$\pm$0.19
Group A (fine-tune only)	Linear probe	90.23$\pm$0.21	32.99$\pm$0.98	72.46$\pm$0.20	42.41$\pm$0.26	76.88$\pm$0.11	59.98$\pm$0.08
	Temporal shuffle	90.48$\pm$0.27	34.42$\pm$0.64	72.66$\pm$0.29	44.88$\pm$0.58	78.14$\pm$0.25	60.82$\pm$0.36
	`[CLS]` → mean pool	90.64$\pm$0.21	33.90$\pm$0.79	73.73$\pm$0.15	46.36$\pm$0.39	78.96$\pm$0.12	61.37$\pm$0.10
	Freeze SWE	90.40$\pm$0.25	33.62$\pm$0.78	73.06$\pm$0.27	45.11$\pm$0.46	78.11$\pm$0.09	60.21$\pm$0.23
	Freeze CSE	90.88$\pm$0.20	35.11$\pm$1.04	73.59$\pm$0.32	45.89$\pm$0.60	78.70$\pm$0.15	61.72$\pm$0.27
Group B (re-pretrained)	− SWE (6L CSE-only)	87.09$\pm$0.46	30.21$\pm$0.73	70.03$\pm$0.41	40.91$\pm$0.44	74.10$\pm$0.30	54.76$\pm$0.56
	− CSE (6L SWE-only)	88.66$\pm$0.16	31.22$\pm$1.11	71.52$\pm$0.30	42.94$\pm$0.63	75.25$\pm$0.19	56.53$\pm$0.18
	− SWE (12L CSE-only)†	88.91$\pm$0.34	32.04$\pm$0.90	71.98$\pm$0.37	42.62$\pm$0.32	75.49$\pm$0.28	58.85$\pm$0.34
	− CSE (12L SWE-only)†	90.02$\pm$0.15	33.45$\pm$1.34	73.48$\pm$0.29	44.39$\pm$0.71	76.58$\pm$0.24	59.37$\pm$0.48
	Sequential (SWE^6 → CSE^6)†	90.92$\pm$0.22	34.08$\pm$1.38	73.00$\pm$0.32	46.68$\pm$0.45	77.59$\pm$0.14	61.06$\pm$0.13

Interleaving outperforms the sequential design. Consistent with the simulation (Section 4.2), the parameter-matched sequential variant (SWE^6 → CSE^6) falls below NEST on every task and metric. Despite being pretrained with MSM, it also falls below NEST (MLM only), indicating that the gap comes from the layer order rather than the pretraining objective. **Both types of blocks are essential, and depth cannot substitute for either.** Removing either encoder degrades all three tasks relative to NEST (MLM only), and doubling the remaining encoder recovers only part of the gap. **NEST relies on encounter order and the `[CLS]` summary.** At fine-tuning, shuffling encounter order or reading out mean-pooled tokens instead of `[CLS]` degrades all three tasks. **MSM adds a consistent gain in MIMIC-IV tasks.** NEST (full) outperforms NEST (MLM only) on every task and metric.

6 Conclusion

We introduced NEST, a hierarchical Transformer for sequence-of-multiset (`SeqSet`) event streams. By interleaving within-set (SWE) and cross-set (CSE) encoders, NEST preserves multiset boundaries and induces structured sparsity, achieving higher throughput than BERT-style baselines while exposing a token-level bypass path that prevents the information loss inherent in sequential SWE-then-CSE designs. Masked Set Modeling adds explicit set-level supervision, yielding directly usable `[CLS]` embeddings without post-hoc pooling. Across MIMIC-IV, a proprietary pediatric EHR, and Instacart NBR, NEST consistently outperforms strong baselines; ablations confirm that gains arise from the interleaved architecture and MSM rather than capacity. NEST is a step toward foundation models that respect, rather than flatten, the structure of real-world event streams.

Limitations. NEST is scoped to representation-learning foundation models, since our goal is patient representation quality rather than benchmarking all model families. Generative trajectory models (Renc et al., 2024; McDermott et al., 2023) optimize a different objective and require a separate evaluation protocol, which we leave to future work. We report discriminative metrics (AUROC/AP) but do not assess calibration, which is essential for clinical risk estimation. Although SeqSet extends beyond healthcare, our cross-domain evidence is limited to a single non-EHR dataset (Instacart); broader validation across event-stream domains (e.g., financial transactions, user-behavior logs) remains an important direction.

AI USE STATEMENT

In this work, we used generative AI tools for cleaning and reformatting dataset, implementing methods. We have not used generative AI tools for help developing methodology or experiments, theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, assisting with translation, designing or providing feedback on research, interpreting results, and providing critical ingredients for proving mathematical claims, and supporting qualitative and generating synthetic data sets, thematic data analysis, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, editing a research paper to improve readability, identifying relevant literature. We have reviewed all AI-assisted work. We verified LLM-generated code for data preprocessing and run files and scripts, we checked all the searched results for information. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The use of ML in healthcare carries significant ethical responsibilities. To ensure data privacy, we exclusively used de-identified MIMIC-IV data accessed via PhysioNet credentialing and IRB-approved proprietary EHRs within secure environments. We acknowledge the risk of algorithmic bias, particularly for under-documented populations due to historical EHR disparities; rigorous fairness evaluations across demographic subgroups are essential prior to any real-world deployment. To mitigate risks from uncertainty and overconfidence, our model is strictly designed as an assistive Clinical Decision Support tool requiring human-in-the-loop oversight. Future deployment would benefit from interpretability analyses such as attention visualization to support clinician trust.

REPRODUCIBILITY STATEMENT

Our code will be publicized upon acceptance. For the submission we provide the anonymous GitHub repository link (https://anonymous.4open.science/r/anonymous-code-submit-444B/) for the access to our code and reproduce the experiment results. The implementation details of using the public datasets and models are all provided in the appendix.

REFERENCES

Mozhdeh Ariannezhad, Sami Jullien, Ming Li, Min Fang, Sebastian Schelter, and Maarten De Rijke. Recanet: A repeat consumption-aware neural network for next basket recommendation in grocery shopping. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 1240–1250, 2022.

Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R Costa-jussà, David Dale, et al. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*, 2024.

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

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in english. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*, 2022.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. Diffcse: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4207–4218, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6894–6910, 2021.

Haoyu He, Markus Flicke, Jan Buchmann, Iryna Gurevych, and Andreas Geiger. Hdt: Hierarchical document transformer. *arXiv preprint arXiv:2407.08330*, 2024.

Haoji Hu and Xiangnan He. Sets2sets: Learning from sequential sets with neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1491–1499, 2019.

Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1071–1080, 2020.

Sihao Hu, Zhen Zhang, Bingqiao Luo, Shengliang Lu, Bingsheng He, and Ling Liu. Bert4eth: A pre-trained transformer for ethereum fraud detection. In *Proceedings of the ACM Web Conference 2023*, pp. 2189–2197, 2023.

Tinglin Huang, Syed Asad Rizvi, Rohan Krishna Thakur, Vimig Socrates, Meili Gupta, David van Dijk, R Andrew Taylor, and Rex Ying. Heart: Learning better representation of ehr data with a heterogeneous relation-aware transformer. *Journal of Biomedical Informatics*, 159:104741, 2024.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Hyewon Jeong, Nassim Oufattole, Matthew Mcdermott, Aparna Balagopalan, Bryan Jangeesingh, Marzyeh Ghassemi, and Collin Stultz. Event-based contrastive learning for medical time series. *arXiv preprint arXiv:2312.10308*, 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.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`, 2022.

Ming Li, Mozhdeh Ariannezhad, Andrew Yates, and Maarten De Rijke. Masked and swapped sequence modeling for next novel basket recommendation in grocery shopping. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 35–46, 2023a.

Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten De Rijke. A next basket recommendation reality check. *ACM Transactions on Information Systems*, 41(4):1–29, 2023b.

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.

Yikuan Li, Mohammad Mamouei, Gholamreza Salimi-Khorshidi, Shishir Rao, Abdelaali Hassaine, Dexter Canoy, Thomas Lukasiewicz, and Kazem Rahimi. Hi-behrt: hierarchical transformer-based model for accurate prediction of clinical events using multimodal longitudinal electronic health records. *IEEE journal of biomedical and health informatics*, 27(2):1106–1117, 2022.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Matthew McDermott, Bret Nestor, Peniel Argaw, and Isaac S Kohane. Event stream gpt: a data pre-processing and modeling library for generative, pre-trained transformers over continuous-time sequences of complex events. *Advances in Neural Information Processing Systems*, 36: 24322–24334, 2023.

Mikkel Odgaard, Kiril Vadimovic Klein, Sanne Møller Thysen, Espen Jimenez-Solem, Martin Sillesen, and Mads Nielsen. Core-behrt: A carefully optimized and rigorously evaluated behrt. *arXiv preprint arXiv:2404.15201*, 2024.

Chao Pang, Xinzhuo Jiang, Krishna S Kalluri, Matthew Spotnitz, RuiJun Chen, Adler Perotte, and Karthik Natarajan. Cehr-bert: Incorporating temporal information from structured ehr data to improve prediction tasks. In *Machine learning for health*, pp. 239–260. PMLR, 2021.

Raphael Poulain and Rahmatollah Beheshti. Graph transformers on EHRs: Better representation improves downstream performance. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pe0Vdv7rsL.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, 2017.

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.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Pawel Renc, Yugang Jia, Anthony E Samir, Jaroslaw Was, Quanzheng Li, David W Bates, and Arkadiusz Sitek. Zero shot health trajectory prediction using transformer. *NPJ digital medicine*, 7 (1):256, 2024.

Ethan Steinberg, Jason Fries, Yizhe Xu, and Nigam Shah. Motor: A time-to-event foundation model for structured medical records. *arXiv preprint arXiv:2301.03150*, 2023.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, 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.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pp. 10524–10533. PMLR, 2020.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 14138–14148, 2021.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489, 2016.

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 Intuition: Bypass Information Preservation

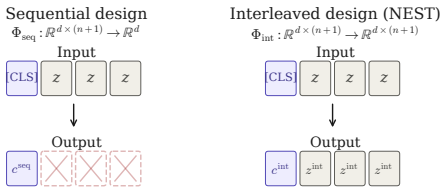

Figure A1: Interleave design preserve token-level information in each NEST layer.

In the sequential design, SWE processes each set on its own, so the `[CLS]` embedding of set i that crosses the SWE–CSE boundary is computed from set i alone, and every later block sees set i only through this embedding. What is kept from set i is therefore the same whatever the other sets are. In the interleaved design, the `[CLS]` of set i receives context from the other sets in each CSE block, and the next SWE block reads the tokens of set i again with this `[CLS]`, so what is kept from set i can depend on the other sets (Figure A1).

A minimal example shows why this matters. Let set i consist of two tokens carrying bits a and b, and let the other sets determine a bit s that selects which one the target uses: $y = a$ if $s = 0$ and $y = b$ if $s = 1$, with a, b, and s independent and uniform. Suppose the summary of set i can keep only one bit. The interleaved design receives s first and then keeps the bit that y needs, so it predicts y without error. In the sequential design, the prediction can use only a one-bit summary of (a, b) formed without s, and any such summary leads to an error of at least $1/4$, however many CSE blocks follow; keeping a, for example, fails half the time when $s = 1$.

The example illustrates the mechanism rather than modeling NEST: a two-bit summary would remove the gap, and a `[CLS]` embedding holds far more than one bit. What carries over is that the sequential design compresses each set without the context, which matters when the summary cannot keep everything the context might ask for. Whether this happens with learned encoders is an empirical question. The simulation in Section 4.2 poses a target of the same kind in continuous form, with the query $\boldsymbol{s}_i$ formed from the other sets, and Section 5 compares the two designs on MIMIC-IV at matched parameter count.

B MSM

B.1 Derivation

Consider a family of probability measures $\{\pi_\theta : \theta \in \Theta\}$ on the finite space $\Omega = \{\omega_1, \ldots, \omega_k\}$. The KL divergence of π_θ from a fixed measure P decomposes as

$$D_{\mathrm{KL}}[P; \pi_\theta] = -H(P) - \sum_{i=1}^{k} P[\omega_i] \log \pi_\theta[\omega_i],$$

so since $H(P)$ is constant in θ,

$$\arg\min_{\theta\in\Theta} D_{\mathrm{KL}}[P; \pi_\theta] = \arg\max_{\theta\in\Theta} \sum_{i=1}^{k} P[\omega_i] \log \pi_\theta[\omega_i].$$

That is, minimizing KL divergence is equivalent to maximizing the negative cross-entropy.

Applying this to the multinomial: let $\mathbf{x} = (x_1, \ldots, x_k)$ be observed counts under $\mathbf{X} \sim \mathrm{Multinomial}(n, \pi_\theta)$, and define the empirical measure $\hat{P}_i = x_i/n$. Since the log-likelihood satisfies

$$\ell(\theta; \mathbf{x}) = C + \sum_{i=1}^{k} x_i \log \theta_i, \quad C \text{ free of } \theta,$$

we have

$$\arg\max_{\theta\in\Theta} \ell(\theta;\mathbf{x}) = \arg\max_{\theta\in\Theta} \sum_{i=1}^{k} \hat{P}_i \log \pi_\theta(i) = \arg\min_{\theta\in\Theta} D_{\mathrm{KL}}[\hat{P};\pi_\theta].$$

Hence MLE is equivalent to minimizing the KL divergence from the empirical measure to the model family.

B.2 Forced Cross-Set Aggregation

The MSM masking protocol (Section 3.5) forces `[CLS]` to encode set-level structure as a direct consequence of the objective, rather than as an emergent side-effect of self-attention. Concretely, with every intra-set token masked under MSM, no within-set signal is available to reconstruct $\hat{P}_i$. The `[CLS]` token must instead aggregate cross-set context through CSE—inferring what tokens belong to a clinical encounter from the patient's broader trajectory. As a result, NEST's `[CLS]` embeddings $\{\boldsymbol{c}_i\}$ are usable directly for set-level downstream tasks without post-hoc pooling, distinguishing NEST from contrastive methods (Gao et al., 2021; Chuang et al., 2022) that provide only implicit pressure for the token to summarize set contents.

B.3 Difference from Segment MLM (SMLM) in language modeling

SMLM vs. MSM: HAT's SMLM (Chalkidis et al., 2022) predicts missing tokens individually, where the cross-entropy loss is operated on the token level. EHR encounters are ordered/unordered bag of tokens. MSM does not predict individual tokens. It is a set-level objective where the `[CLS]` token predicts the entire token distribution of the multiset at once, which is derived via multinomial likelihood. SMLM reconstructs tokens in the segment individually; MSM reconstructs an set distribution using the `[CLS]` token.

B.4 Ablation Study: What Does MSM Provide?

We ablate the MSM objective to understand its contribution to NEST pretraining. Specifically, we ask: (1) Does MSM improve set-level prediction? (2) Does the learned representation transfer beyond the task-specific head? We perform experiments on both Instacart and proprietary EHR. We pretrain three small (3-layer) NEST variants for 10 epochs: MSM-only, MLM-only, and MLM+MSM. We measure set prediction performance using a weight-tied linear layer (Press & Wolf, 2017) and MSM task head used in the pretraining.

Table B1: Zero-shot readout (Recall@10 / NDCG@10). "-" marks a head that was never trained under that objective, so the readout is undefined there.

Pretraining	`1st-[MASK]` + MLM head	`[CLS]` + MSM head
Proprietary EHR		
MLM-only	53.23 / 49.85	—
MLM + MSM	53.15 / 51.03	64.26 / 61.58
MSM-only	—	64.62 / 61.93
Instacart		
MLM-only	29.45 / 35.96	—
MLM + MSM	26.62 / 32.55	24.92 / 24.46
MSM-only	—	20.74 / 21.46

This probe predicts the last set from preceding context.

MSM-only does learn transferable representations. Read out through its own head it performs reasonably well on the set imputation. MSM's gradient reaches the encoder only through `[CLS]`, so under MSM-only the token embeddings never get direct supervision and the MLM readout has nothing trained to read from. MLM supervises the token path directly and stands on its own. Each single-objective model works through the path its objective trains.

C MULTISET PREDICTION

During pretraining, we employ a separate MSM task head for set prediction. However, this setup may overestimate representation quality, as set-level information can be largely absorbed by the MSM-task head while the FM backbone remains insufficiently optimized—in other words, the MSM-task head may perform most of the heavy lifting. To mitigate this issue, we adopt a more conservative strategy when evaluating FM representation quality by computing the missing tokens for a masked set without relying on the MSM head. Specifically, analogous to MLM, we pass the output representation through the shared, weight-tied embedding projection to compute item logits, then select the top-K scoring items to form the predicted basket.

As the predicted set is formed from the top-K predicted token logits, we evaluate set prediction using Recall@K and NDCG@K (normalized discounted cumulative gain). Recall@K measures the fraction of ground-truth basket items recovered within the top-K predictions, while NDCG@K rewards models that rank ground-truth items more confidently toward the top of that list of logits.

$$\text{Recall@}K(u) = \frac{|T_u \cap P_u|}{|T_u|} \tag{5}$$

$$\text{NDCG@}K(u) = \frac{\sum_{k=1}^{K} p_k / \log_2(k+1)}{\sum_{k=1}^{\min(K,|T_u|)} 1/\log_2(k+1)} \tag{6}$$

where T_u and P_u are the ground truth set (which need not be in a size of K) and predicted set with K recommended items for subject u and $p_k = \mathbb{I}\{P_u^{(k)} \in T_u\}$. Recall@$K$ quantifies model's ability to find the relevant items, and NDCG@K measures model's ability to push all correct tokens to the front of the recalled tokens.

D SIMULATION STUDIES

D.1 COMPUTING EFFICIENCY

We calculate the throughput over 100 random batches across five random seeds. We reported pooled standard errors.

$$\sigma_{\text{pooled}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} s_i^2 + \text{Var}(\bar{x}_i)}$$

D.2 INTERLEAVE

Let $\mathcal{C} \in \mathbb{R}^{K \times d_{\text{model}}}$ be a token content codebook with each row drawn i.i.d. from $\mathcal{N}(0, I_{d_{\text{model}}})$. For each set i, draw $\boldsymbol{\pi}^{(i)} \sim \text{Dir}(\boldsymbol{\alpha})$, sample $k_{ij}|\boldsymbol{\pi} \overset{\text{i.i.d.}}{\sim} \text{Cat}(\boldsymbol{\pi}^{(i)})$, and set

$$\boldsymbol{x}_{ij} = \mathcal{C}[k_{ij}] + \boldsymbol{\epsilon}_{ij}, \quad \boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(0, \sigma^2 I_{d_{\text{model}}}),$$

so each token combines codebook content with Gaussian noise. Then, we define the cross-set summary and synergy-dependent outcome as follows:

$$\boldsymbol{s}_i = \frac{1}{(T-1)\sqrt{n}} \sum_{i' \neq i} \sum_{j} \boldsymbol{x}_{i'j}, \qquad y = \sum_{ij} \text{softmax}_j\big(\langle \boldsymbol{x}_{ij}, \boldsymbol{s}_i \rangle\big) \cdot \langle \boldsymbol{x}_{ij}, \boldsymbol{s}_i \rangle. \tag{7}$$

where the softmax scores are driven both by the token content but also noise with controllable scales. Intuitively, y counts repeated codes. For each token in set t, the inner product with the average of the other sets' tokens is ≈ 1 when the two share the same code and ≈ 0 otherwise, so only codes that appear in both this set and the others can produce a large logit. After the softmax over tokens, y is therefore, up to a fixed $1/((T-1)\sqrt{n})$ factor, a count of the strongest repeat across the window. This count is high only when two conditions hold jointly: the per-set distribution $\pi^{(t)}$ is spiky, concentrating several of the n tokens on a single code, and the sets draw from overlapping regions of the codebook. Conversely, when codes do not repeat (e.g., disjoint sets whose tokens are mutually near-orthogonal) every cross term vanishes, the softmax goes flat, and $y \approx 0$.

In the simulation experiments, we use two different kinds of $\boldsymbol{\alpha}$: (a) $\boldsymbol{\alpha} = (\alpha_0, ..., \alpha_0)$ that results in uniform token distribution, and (b). $\boldsymbol{\alpha} = (r\alpha, r\alpha^2, ..., r\alpha^K)$ that results in nonuniform token distribution. We also conduct experiments under varied level of noise by changing σ^2. In this simulation study, we evaluated two distinct configurations of the SWE and CSE blocks: for example, an interleaved arrangement (`SWE-CSE-SWE-CSE`), which is the NEST architecture, and an ad hoc sequential arrangement (`SWE-SWE-CSE-CSE`).

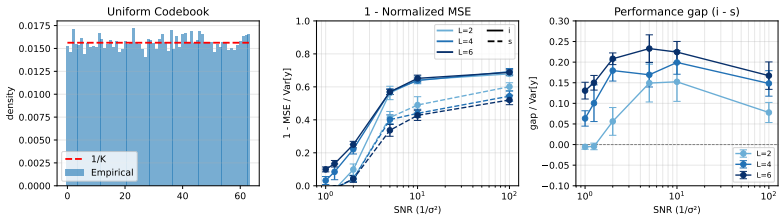

Figure D1: results for uniform codebook

The simulation uses end-to-end supervised training without self-supervised pretraining, so we adopt Pooling by Multihead Attention (Lee et al., 2019) for set-level pooling as a proof of concept, which is architecturally guaranteed to aggregate token information.

E Real World Datasets

E.1 Instacart

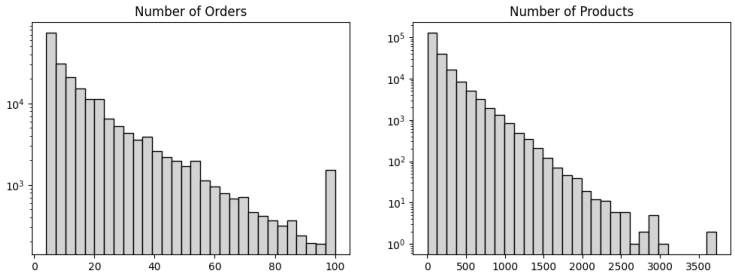

Figure E1: Distributions for # multisets and # tokens per user.

Instead of using a random split, we follow the official split. For each user in the training set, the chronological history of baskets is used for training. For final inference evaluation, the last basket in the test set is used as the outcome, while the preceding basket history in the test set is used as the input.

E.2 MIMIC-IV v3.1 - hosp

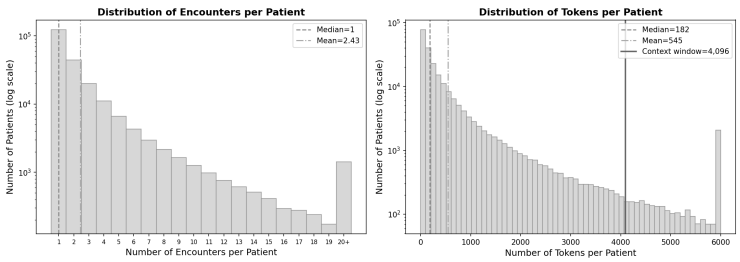

Figure E2: Distributions for # multisets and # tokens per patient.

Tokenization. Following ETHOS (Renc et al., 2024), we construct four clinical event types with standardized medical codes:

- **Diagnoses (`DX`):** ICD-9-CM codes are converted to ICD-10-CM using CMS General Equivalence Mappings (GEM). When one ICD-9 maps to multiple ICD-10 codes, we select the shortest following ETHOS. Unmappable codes (marked `NoDx` in GEM) are excluded. Format: `DX:{icd10}`.
- **Procedures (`PR`):** ICD-9-CM procedure codes are converted to ICD-10-PCS with the same shortest-match strategy. Unmappable codes (`NoPCS`) are excluded. Format: `PR:{icd10pcs}`.
- **Medications (`MED`):** We extract administered medications from `emar` (filtering `event_txt='Administered'`). Drug names are mapped to ATC codes via a lookup table constructed from `prescriptions`: the GSN (Generic Sequence Number) field is mapped to ATC using external GSN-ATC mappings. Format: `MED:{atc}`.
- **Labs (`LAB`):** We select the 200 most frequent lab items. Continuous values are discretized into deciles (Q1–Q10) using population-level quantile thresholds. Format: `LAB:{itemid}_Q{1-10}`. This approach integrates numerical information into the modeling process.

Event timestamps. We timestamp lab results with the time the result becomes available in the record, and medications with the time they are administered. For lab results, we use `storetime`, when the result is reported, instead of `charttime`, when the specimen is collected. Lab results without a `storetime` are discarded. For medications, we use the administration time recorded at the bedside (`charttime` of the `emar` record). These records are entered with little delay (median 1 minute; 99% within 63 minutes). Procedure codes only have a date (`chartdate`), so we place them at the end of that day (23:59:59), the latest possible time on that date. Procedures dated before the day of admission are placed at the start of the admission, after its diagnoses. Diagnoses have no timestamps in MIMIC-IV and are anchored at the start of the admission, following ETHOS's convention of using `admittime` as the diagnosis timestamp. For tasks that predict before discharge, the diagnosis and procedure codes of the target admission are removed (Appendix E.2.1). Lab results and medications linked to an admission (`hadm_id`) are kept if they are collected or administered no earlier than 24 hours before `admittime`. A lab result without a `hadm_id` is assigned to the nearest admission that starts within 24 hours after it is collected. Finally, timestamped events that fall after the discharge time (`dischtime`) of their own admission are removed, so a lab result reported after the patient leaves the hospital never enters that admission.

Hierarchical structure. The principal NEST-specific adaptation is the *hierarchical SeqSet* formulation: rather than ETHOS's flat 1D timeline with discrete time-interval separator tokens, NEST groups events into a sequence of multisets, where each multiset corresponds to a single hospital admission. Within each multiset, events are ordered by the timestamps defined above, expressed as offsets in hours from `admittime`; ties are broken by `seq_num` (for diagnoses and procedures, their recorded priority) and then by code. Across multisets, the number of days since the first admission in the input instantiates the inter-set position scalar $\texttt{pos}_i$ (Section 3.3) for CSE's RoPE.

E.2.1 Downstream Tasks

We evaluate NEST on five downstream tasks, covering clinical outcome prediction, diagnosis classification, and future event prediction. All tasks use patient-level 70/15/15 train/validation/test splits to prevent information leakage between patients.

Prediction Mechanism. For all downstream tasks, the pre-trained encoder processes the patient's admission sequence and produces contextualized representations. Each segment is prepended with a `[CLS]` token. For classification, we extract the `[CLS]` representation from the last valid segment and pass it through a two-layer MLP (with GELU activation and dropout) to produce logits. Binary and multi-class tasks use cross-entropy loss; multi-label tasks (ICD category) use binary cross-entropy loss.

Task Definitions.

- **Inpatient Mortality.** Binary classification predicting whether a patient dies during the hospital stay. Following prior work, we make the prediction 24 hours after admission, using only data available by then. Diagnosis and procedure codes from the target admission are excluded. Evaluated using AUROC and AUPRC.
- **30-Day Readmission.** Binary classification predicting whether a patient is readmitted within 30 days after discharge. We include every admission that ends in a live discharge, and label it positive if the patient's next admission begins within 30 days. Prediction is made at discharge time using full admission data (including diagnosis and procedure codes). Evaluated using AUROC and AUPRC.
- **Prolonged Length of Stay.** Binary classification predicting whether the hospital stay exceeds 7 days. We make the prediction 48 hours after admission, using only data available by then. Diagnosis and procedure codes from the target admission are excluded. Evaluated using AUROC and AUPRC.
- **ICD Chapter Classification.** Multi-class classification predicting the primary diagnosis ICD chapter (based on the `seq_num=1` diagnosis code). ICD-10-CM codes are mapped to 22 chapters; ICD-9-CM codes are mapped to corresponding chapter equivalents. Prediction is made at discharge, with target admission diagnoses excluded. Evaluated using macro-averaged AUROC and AUPRC.
- **ICD Category Multi-label Classification.** Multi-label classification predicting all 3-character ICD categories present in the admission. We retain categories that appear at least 100 times across all diagnosis records. Labels are represented as multi-hot vectors over the category vocabulary. Prediction is made at discharge with target diagnoses excluded. Evaluated using macro-averaged AUROC and AUPRC.

Data Leakage Prevention. We build the inputs of all tasks with the following rules, which are the same for NEST and all baselines. For each task and seed, all models are fine-tuned and evaluated on the same patient-level split.

- **Event timing.** All events are timestamped as described in Appendix E.2. When the target admission is truncated at the prediction time, it only keeps the lab results that have been reported and the medications that have been administered by then.
- **Prediction windows.** For mortality and prolonged LoS, we use only the events of the target admission available by 24h and 48h after admission, respectively. Readmission and the two ICD tasks are predicted at discharge, so their window covers the whole target admission.
- **Target-admission codes.** Diagnosis (`DX`) and procedure (`PR`) codes are assigned when the admission is coded at discharge, and diagnoses have no timestamps. We therefore remove both from the target admission for mortality and prolonged LoS. For the two ICD tasks, the labels come from the diagnosis codes of the target admission, so these codes are removed. Readmission keeps both diagnosis and procedure codes.
- **Cohort eligibility.** An admission that has already ended before the prediction time is not a prediction target. Mortality and prolonged LoS only include admissions where the patient is still in hospital, neither discharged nor deceased, at 24h and 48h after admission, respectively. Readmission only includes admissions that end in a live discharge. Whether an admission belongs to one of these cohorts is thus decided by information available at the prediction time.
- **Patient history.** The patient's earlier admissions come before the target admission in the input, including admissions that are not prediction targets themselves. Later admissions are not included.

Each input thus reflects the patient's record at the prediction time of the task, mirroring what a clinical decision-support system could observe at the moment of prediction.

Evaluation Metrics. Binary classification tasks use AUROC and AUPRC as primary metrics. Multi-class and multi-label tasks use macro-averaged AUROC and AUPRC, computed via one-vs-rest per class then unweighted averaging.

Table E1: Summary of downstream task configurations. Removed codes are withheld from the target admission only.

Task	Type	Prediction Time	Removed Codes	Primary Metric
Mortality	Binary	Admission + 24h	DX, PR	AUROC / AUPRC
30-Day Readmission	Binary	Discharge	None	AUROC / AUPRC
Prolonged LoS	Binary	Admission + 48h	DX, PR	AUROC / AUPRC
ICD Chapter	Multi-class	Discharge	DX	Macro AUROC / AUPRC
ICD Category	Multi-label	Discharge	DX	Macro AUROC / AUPRC

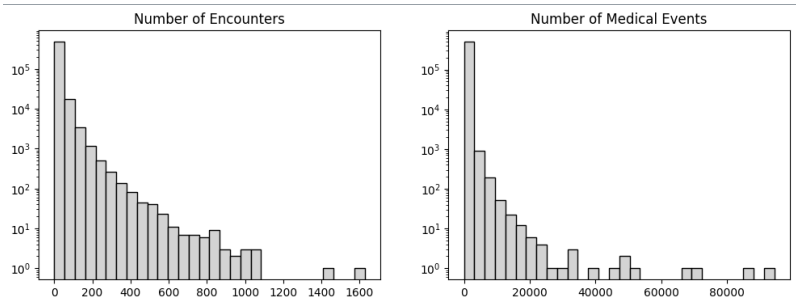

Figure E3: Distributions for # multisets and # tokens per patient.

E.2.2 Ablation studies explanation

Both encoders are essential, and depth cannot substitute for either. Removing CSE degrades all three tasks substantially; doubling SWE depth to match NEST's parameter count recovers only part of the gap. Replacing SWE's parametric attention with parameter-free mean pooling incurs an even larger drop, exceeding the parameter-matched SWE-only variant on every metric. Depth alone cannot substitute on either side.

Interleaving matters beyond having both encoders. The sequential variant keeps both encoders and NEST's parameter count, but stacks all SWE blocks before all CSE blocks. It falls below NEST on every task and metric, and even below NEST (MLM only) despite being pretrained with MSM, indicating that the gap comes from the layer order rather than the pretraining objective.

CSE relies on the temporal order of encounters. Shuffling encounter order at fine-tuning time degrades all three tasks, indicating that CSE's downstream contribution depends on the temporal order it encodes.

The `[CLS]` pathway carries set-level information that mean pooling does not recover. Replacing `[CLS]` with mean pooling at fine-tuning time degrades all tasks.

SWE representations require more task-specific adaptation than CSE representations. Freezing SWE hurts more than freezing CSE on every task: within-encounter representations carry more of the task-specific signal that fine-tuning must adapt, while pretrained cross-encounter temporal patterns transfer with less adjustment.

Together, these ablations show that NEST's gains arise from its architectural commitments rather than capacity.

E.3 Proprietary EHR

The train-test split in the downstream task cohorts follows the deterministic birth-date–based split of the pretrain data. Patients born on the 17th–24th of each month are assigned to the test set and the remainder to the training set.

E.3.1 ETL PIPELINE

We implemented an ETL data processing pipeline to extract multi-source, heterogeneous EHR tables from a clinical Oracle database conforming to the PCORnet Common Data Model (CDM). The ETL process is organized around patients' longitudinal clinical histories and retrieves structured clinical events from patient demographics, diagnoses, procedures, laboratory measurements, medication administrations and prescriptions, vital signs, and immunizations. All records are joined, filtered, and indexed at both the patient and encounter levels. Clinical records from different domains are subsequently normalized into a shared base schema, consisting of standardized patient and encounter identifiers, a unified event timestamp, and event-type–specific code fields along with the required numerical attributes, enabling consistent event-level representation across domains for downstream sequence modeling.

E.3.2 DOWNSTREAM COHORTS

Recurrent Acute Otitis Media (rAOM) rAOM, or recurrent ear infection, is a common medical condition in early childhood. While most children will experience an ear infection during their first years of life, approximately 20-30% of the pediatric population will experience recurrent ear infections defined as 3 episodes within a 6 month period or 4 episodes within a year period. Children who experience rAOM often require surgical intervention in the form of ear tube placement. We sought to develop a predictive model for the probability a first AOM develops into rAOM.

We identified a cohort of 3,852 children with a first AOM. Of these, 22.0% transitioned into rAOM. We used the same embeddings framework as above to process prenatal, birth and developmental clinical data. Since our eligible cohort had reached age 4 — an age at which rAOM is unlikely to developed — we modeled the binary diagnosis indicator as the outcome. Our training and testing sizes were 2,822 and 1,030, respectively.

Autism Spectrum Disorder (ASD) ASD is a neuro-developmental condition that typically presents within the first years of life. The mean age of the diagnosis is around 5 years. Children are typically screened during their 18-month well-child visit and referred for confirmatory diagnosis. Previous works has shown that early medical conditions (e.g., gastrointestinal disease) can serve as early indicators of a future ASD diagnosis. Our team has been focusing on developing automated early screening tools for ASD.

We identified a cohort of 43,945 children who had a well-child visit at our institution between 12 and 24 months of age. We used the well-child visits closest to 18 months as the index encounter. We required two distinct medical encounters with an ICD-10 for ASD to label an ASD case. Across our cohort's follow-up, 2.05% were diagnosed with ASD. Our training and testing sizes were 35,000 and 8,945, respectively.

30-day Hospital Readmission Hospital readmission is a commonly used measure of short-term healthcare utilization and quality of care in pediatric populations. We focus on 30-day hospital readmission following emergency inpatient or inpatient hospitalization encounters among children.

We identified a cohort of 26,617 children who had at least one emergency inpatient (EI) or inpatient hospitalization (IP) encounter at our institution between January 1, 2016 and age 18. After excluding birth-related encounters, we used all qualifying EI or IP encounters with a discharge disposition of Home/Self Care (HO) or Home Health (HH) as index encounters, allowing multiple index encounters per patient. We defined the outcome as 30-day readmission, requiring the occurrence of a subsequent Emergency Department (ED), EI, or IP encounter within 30 days following discharge from the index encounter. The final cohort consisted of 45,755 index encounters, of which 19.1% were followed by a 30-day readmission. Our training and testing sizes were 33,683 and 12,072, respectively.

F Implementation Details

F.1 Instacart

F.1.1 Data and Evaluation Protocol

All systems use the same fixed user-level partition of the Instacart corpus. The corpus test users form the test fold, while the training users are split 90/10 into training and validation sets. A basket corresponds to one order rather than one calendar day. For each user, the final basket is predicted from all preceding baskets; users with fewer than two baskets are excluded. Recall and NDCG use binary relevance against the full target basket.

Table F1: Data and evaluation protocol.

Corpus users	131,209 training / 75,000 test
Training/validation users	118,089 / 13,120; 90/10 split with a fixed seed
Item vocabulary	∼45,000 products; special tokens excluded
Prediction target	Final basket for each user
History	All preceding baskets; minimum history of two baskets
Metrics	Recall@K and NDCG@K, for $K \in \{10, 20\}$
Model selection	Validation NDCG@10; best weights restored

F.1.2 Pretrained Encoders & Baselines

The two encoders share the same set of hyperparameters and optimization procedure, differing only in the input geometry and pretraining objectives, which can directly perform zero-shot recommendations.

Table F2: Shared encoder configuration and encoder-specific objectives.

	NEST	**BERT4Rec**
Architectural parameters	768-dim, 12 heads, 6 blocks, 2048-dim SwiGLU FFN	
Normalisation / dropout	LayerNorm / 0.0	
Embeddings	Tied input–output embeddings; Time2Vec temporal features	
Optimisation	AdamW, $5e{-5}$, batch size 16, 30 epochs	
Input geometry	64 baskets × 32 items	2,048 tokens with basket sentinels
Attention	Within basket, then across baskets	Full attention with basket-index RoPE
Item objective	Masked item modelling, rate 0.20	Masked item modelling, rate 0.20
Basket objective	Masked basket modelling against held-out item distribution; rate 0.40	None
Objective weights	1 for both MLM & MSM	—

The downstream ReCANet module is fixed across all arms. Each candidate item is scored from the user and item representations together with the item's recent purchase-interval sequence using a two-layer LSTM. Candidates are all items purchased at any point in the user's history, and training minimizes binary cross-entropy. All arms use a hidden dimension of 32, a five-basket interval window, AdamW with learning rate 10^{-4}, and at most 10 epochs.

Table F3: ReCANet recommender configurations.

	Pretrained NEST	**Pretrained BERT4Rec**	**From scratch**
Encoder	Frozen NEST	Frozen BERT4Rec	None
User representation	Mean over basket representations	Mean over basket sentinels	Masked mean of item embeddings
Item embeddings	Encoder, $768 \to 128 \to 32$	Encoder, $768 \to 128 \to 32$	Learned, 32-dimensional
Input geometry	64×32	2,048 tokens	64×32
Batch size (users)	16	16	64

In the pretrained arms, the encoder is frozen and both user and item representations are projected into the scoring space. The prediction-point cap bounds the batch size without permanently discarding supervision. The from-scratch arm learns its own item table and omits the original per-user embedding table: because every test user is unseen, that table would remain at random initialisation.

Table F4: Baseline configurations.

	TIFU-KNN	**Sets2Sets**
Core configuration	3 temporal groups; within/across-group decay 0.9/0.7; 900 neighbours; interpolation weight 0.9; retain top 100	Basket-sum embeddings; one recurrent layer; hidden dimension 32; history capped at 100 baskets
Optimisation	Not applicable	Adam, $\beta = (0.9, 0.98)$, $\epsilon = 10^{-11}$; 10^{-4} learning rate; batch size 128
Training schedule	Not applicable	At most 100 epochs; patience 8; ranking-loss weight 10
Tuning / components	Neighbour count and interpolation weight tuned on validation; remaining values from the original Instacart optimum	Diagonal repeat gate, initialised to the identity

F.2 MIMIC-IV

This section reports implementation settings used in pretraining, downstream fine-tuning, and ablation experiments. To ensure fair comparison, we standardize key hyperparameters across models where possible: all MLM-based pretraining uses mask ratio `0.20`, and all downstream fine-tuning uses learning rate `1e-5`. Per-model differences in weight decay and pretraining objective reflect each baseline's original protocol where deviation would alter the model's identity; pretraining learning-rate schedules are listed in Table F6. Vocabulary size is read at runtime from the tokenizer; in our checkpoints this is `20,275` (CEHR-BERT adds 18 extra special tokens for `[VS]`, `[VE]`, `W0-W3`, `M1-M11`, `LT`, yielding `20,293`; HEART adds four type-specific mask tokens `[MASK_DX]`, `[MASK_PR]`, `[MASK_MED]`, `[MASK_LAB]`, yielding `20,279`).

Table F5: Backbone architecture and pretraining setup, including objective type, masking policy, warmup, and loss weighting in reported runs. Scale counts the parameters trained during pretraining.

Model	Objective	Key pretraining settings	Core architecture	Scale
NEST (default)	MLM + MSM	token mask `0.20`, encounter mask `0.40`, loss weights MLM = MSM = `1.0`, no warmup	`d_model=768`, `n_blocks=6`, `n_heads=12`, `d_ff=2048`, SWE+RoPE, T2V (`scale=1.0`), `max_seg=8`, `max_seq_len=512`, MLM head tied to the token embeddings	∼121M
NEST (MLM only)	MLM only	token mask `0.20`, no warmup	Same backbone and tied MLM head as default NEST	∼101M
CORE-BEHRT	MLM	mask `0.20`, no warmup	`768/12/12/2048`, `max_seq_len=2048`	∼116M
HEART	Missing entity prediction (MEP) + anomaly detection	MEP drop rate `0.70` per entity type, anomaly rate `0.05`, loss weights MEP = `1.0`, anomaly = `0.25`, no warmup	`768/9/12/2048`, per-type heterogeneous edge module, entity- and encounter-level attention, `max_visits=20`, `max_per_type=25`	∼122M
MOTOR	TTE	`n_tasks=512`, `n_time_bins=7`, `final_layer_size=32`, no warmup	Encoder `768/12/12/2048`	∼101M
CEHR-BERT	MLM	mask `0.20`, warmup ratio `0.06`	`hidden=768`, `layers=12`, `heads=12`, `intermediate=3072`, ATT tokens, `n_time_embd=16`	∼117M
GT-BEHRT	NAM → MNP+VTP	NAM mask `0.20` (10 epochs), VTP mask `0.50`, warmup ratio `0.1`	`hidden=780`, `graph_layers=3`, `bert_layers=12`, `heads=12`	∼102M
Hi-BEHRT (BYOL)	BYOL	segment mask `0.20`, momentum `0.996`, warmup ratio `0.1`	`d_model=768/9+10/12/2048`, extractor/aggregator hierarchy, `window=50`, `stride=30`, `t2v_dim=64`	∼121M (BYOL: trainable parameters; target network is an EMA copy)

Table F6: Optimization protocol for pretraining. NEST, CORE-BEHRT, HEART and MOTOR use pre-norm Transformer blocks, which do not require learning-rate warmup (Xiong et al., 2020), and are pretrained at a constant LR. The post-norm CEHR-BERT and Hi-BEHRT use linear warmup followed by cosine decay for training stability; GT-BEHRT uses linear warmup followed by linear decay.

Model	LR	Scheduler	Epochs	Effective batch size
NEST	5e−5	none	30	72
CORE-BEHRT	5e−5	none	30	16
HEART	5e−5	none	30	32
MOTOR	5e−5	none	30	32
CEHR-BERT	3e−5	linear warmup (0.06) + cosine decay	30	64
GT-BEHRT	5e−5	linear warmup (0.1) + linear decay	NAM 10 + main 30	48
Hi-BEHRT (BYOL)	5e−5	linear warmup (0.1) + cosine decay	30	32

Table F8: Task and seed protocol.

Category	Scope	Seeds
Downstream tasks	mortality, readmission_30d, prolonged_los, icd_chapter, icd_category_multilabel	7, 19, 123, 2026, 3507
Ablation reporting protocol	Group A + Group B results are averaged over three random seeds	7, 19, 123

Table F7: Downstream fine-tuning protocol used across all models and tasks.

Setting	Value
Optimizer	AdamW (weight decay: 0.001 for NEST, 0.01 for baselines)
Learning rate	1e−5
Scheduler	linear warmup + cosine decay
Epochs	30
Classification head	two-layer MLP on top of each model's patient representation (for NEST, the last-segment [CLS], or mean-pooled non-[CLS] tokens for the [CLS] → mean pool ablation), with model-specific activation: GELU for NEST, CORE-BEHRT, MOTOR, CEHR-BERT, GT-BEHRT, and HEART; Tanh for Hi-BEHRT
Loss function	cross-entropy for binary and multi-class tasks; binary cross-entropy with logits for multi-label tasks
Early stopping metric	validation AUPRC for binary tasks and validation AUROC for multi-class and multi-label tasks (all models)
Backbone treatment	full fine-tuning by default; ablation variants (Freeze SWE, Freeze CSE, Linear probe) freeze specified components
Data split and seed control	Patient-level splits. Fine-tuning uses a 70/15/15 (train/val/test) split for each task and seed, shared by all models; pretraining uses a separate fixed split.
Early stopping patience	10 for pretraining and 3 for downstream fine-tuning.
Mixed precision detail	CUDA AMP with FP16 (autocast + GradScaler); HEART uses bf16 autocast.

Hardware. All runs use one NVIDIA H200 GPU, 10 CPU cores, and 36GB host memory per job.

G Ablation Setup

G.1 Group A: finetune or probing

Group A keeps the NEST (MLM+MSM) checkpoint fixed and intervenes only at fine-tuning, isolating individual inductive biases:

- **Linear probe:** both encoders frozen; only the classification head is trained. Lies below both Freeze variants on every metric, confirming that fine-tuning some backbone component is necessary for full task adaptation.
- **Temporal shuffle:** encounter order randomly permuted before fine-tuning.
- **`[CLS]` → mean pool:** downstream classifier reads mean-pooled non-`[CLS]` tokens instead of `[CLS]`.
- **Freeze SWE / Freeze CSE:** one encoder frozen, the other trainable.

G.2 Group B: re-pretrain

Group B weakens internal NEST components and re-pretrains from scratch under matched data and optimization budget. The variants are:

- **− CSE (6L/12L SWE-only):** CSE blocks removed entirely. The 12L variant matches the NEST encoder in parameter count.
- **− SWE (6L/12L CSE-only):** SWE's attention and FFN are replaced with parameter-free mean pooling over non-`[CLS]` tokens, with CSE preserved. A broadcast residual propagates cross-encounter context back to all token positions.[2] The 12L variant matches the NEST encoder in parameter count.
- **Sequential (SWE^6 → CSE^6):** all six SWE blocks are applied before all six CSE blocks. At the boundary, only the `[CLS]` tokens are passed to CSE, and the other token states are discarded. It is pretrained with MLM and MSM like NEST, and matches the NEST encoder in parameter count.

H Attention Pattern Visualization

H.1 Hierarchical Pattern

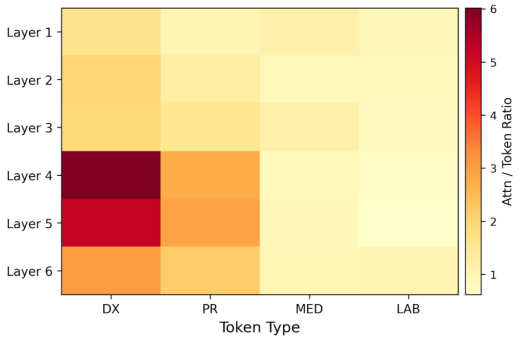

Figure H1: NEST captures hierarchical structure via within-set attention in the SWE module. The heatmap displays the deviation of `[CLS]` attention weights from a uniform baseline across context tokens using the MIMIC-IV data. `DX`: diagnosis; `PR`: procedure; `MED`: medication; `LAB`: lab measures.

[2] A direct "remove SWE" variant is infeasible since CSE operates on `[CLS]` tokens produced by SWE; mean pooling is the closest empirical proxy.

H.2 TEMPORAL PATTERN

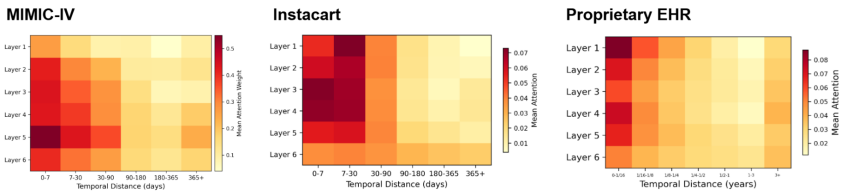

Figure H2: CSE learns temporal recency pattern among the `[CLS]` tokens.