SoAR-SSM: Sobolev-Regularized State Space Models with ARMA Neural Networks for Temporal Hypergraph Reasoning In Health Care

Anonymous authors
Paper under double-blind review

Abstract

Medical knowledge graphs (MKGs) provide a structured mechanism for incorporating clinical knowledge into electronic health record (EHR) modeling. However, existing MKG-based approaches remain limited in three important aspects. First, they predominantly encode pairwise relations or collapse temporal records into static graphs, thereby overlooking higher-order clinical interactions and their evolution across visits. Second, graph representations are commonly learned using message-passing operators that exhibit a strong low-frequency bias and are susceptible to oversmoothing, potentially suppressing localized and clinically salient variations distributed across different spectral regimes. Third, existing approaches often lack an explicit mechanism for preserving long-range patient history. To address these limitations, we propose `SoAR-SSM`, a Sobolev-regularized State Space Model with Auto Regressive Moving Average (ARMA) hypergraph neural networks for frequency-aware temporal hypergraph reasoning. `SoAR-SSM` represents each clinical visit as a knowledge-infused hypergraph to capture higher-order interactions among conditions, procedures, and medications. We then derive a Sobolev-regularized temporal SSM from first principles, yielding a topology-aware spectral operator that governs how structural information is injected into long-range temporal memory. To obtain a more expressive spectral realization, we approximate this operator using parallel ARMA hypergraph neural networks, whose rational responses enable the model to capture complementary information across multiple frequency regimes while mitigating oversmoothing. We evaluate `SoAR-SSM` on MIMIC-III and MIMIC-IV across multiple clinical prediction tasks. The results demonstrate the effectiveness of jointly modeling higher-order clinical interactions, multi-frequency structural information, and long-range temporal dependencies, while maintaining parameter-efficient EHR representation learning.

1 Introduction

Knowledge graphs (KGs) offer a structured and relational representation of domain-specific knowledge, encoding the semantic relationships between the entities (Hogan et al., 2021; Gao et al., 2025). Specifically, KGs augmented with large language models (LLMs) as an inductive bias are increasingly becoming popular in enhancing the reasoning and generalization capabilities of LLMs, particularly in domains where factual consistency and interpretability are critical (Wu et al., 2024; 2025; Jiang et al., 2023). In the medical domain, medical knowledge graphs (MKGs) encode relationship between the medical concepts-such as diagnoses, procedures, and medications (Aldughayfiq et al., 2023; Shirai et al., 2021). Integrating MKGs with electronic health record (EHR) data has demonstrated improved performance and interpretability on downstream clinical tasks such as mortality prediction, drug recommendation, to name a few (Jiang et al., 2025; 2024; Gao et al., 2025).

Machine learning methods ranging from representation-learning approaches (Choi et al., 2016b;a; 2017; Ma et al., 2018; 2020; Zhang et al., 2021; Jiang et al., 2024) to large language models (LLMs) (Jiang et al., 2023; Wu et al., 2024; 2025) are increasingly being adopted for clinical decision support. Specifically, learning based methods are effective at capturing latent patterns in EHR data while remaining relatively parameter- and cost-efficient, whereas LLM-based approaches offer improved interpretability through explicit reasoning. Despite this progress, three key challenges

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.

remain central to temporal EHR modeling and learning based methods. Recent knowledge-enhanced methods partially address these challenges by leveraging a personalized MKGs (Jiang et al., 2025; 2024). However, conventional MKGs predominantly encode pairwise relations, limiting their ability to represent higher-order interactions among co-occurring concepts (conditions, procedures, and medications) and are crucial as patient's clinical state is often characterized by combinations of such concepts whose significance evolves over time. For example, severe respiratory failure may be jointly reflected by acute respiratory distress syndrome, hypoxemia, ventilation, and associated medications, with their clinical relevance depending not only on their co-occurrence but also on how they emerge.

Hypergraphs provide a natural abstraction for modeling such higher-order interactions among clinically related concepts and have been explored for EHR modeling (Xu et al., 2023; Xie et al., 2025). Nevertheless, existing hypergraph based approaches represent an entire visit as a single hyperedge or collapse temporal visits into a static structure, limiting their ability to jointly model rich intra-visit structure and its temporal evolution. Beyond this structural-temporal limitation, most graph and hypergraph representation-learning methods rely on message-passing or finite impulse response filters to obtain representations. They typically model low-frequency components and are susceptible to oversmoothing. This is particularly restrictive for EHR data, where both persistent conditions and localized can be clinically important. A model should therefore preserve information distributed across different spectral regimes while retaining its influence over long temporal horizons. Together with the substantial computational overhead of several knowledge-enhanced approaches (see Fig. 1), motivate a unified and parameter-efficient framework that jointly models temporal higher-order clinical interactions, multi-frequency structural information, and long-range temporal dependencies.

In this work, we introduce `SoAR-SSM`, a parameter-efficient framework for jointly modeling higher-order clinical interactions, multi-frequency structural information, and long-range temporal dependencies in EHR data. `SoAR-SSM` first represents each patient's temporal record as a sequence of knowledge-infused visit-level hypergraphs, where semantically related concepts are connected through hyperedges derived from a global hyperknowledge graph. This representation captures higher-order clinical structure within each visit while preserving its evolution across visits. Building on this temporal hypergraph representations, we then derive a hypergraph-aware temporal state space model from first principles using a Sobolev regularized Higher order polynomial projection operator (HiPPO) approximation objective. Specifically, the derived memory coefficients admit closed form solution as Sobolev-filtered HiPPO representations, yielding a frequency-dependent state space dynamics. Further, for an efficient spectral realization, we implement the Sobolev filters with a proposed hypergraph ARMA neural networks, and are capable of learning complementary spectral responses. The resulting model therefore combines multi-frequency hypergraph representations, with state-space dynamics to overcome over-smoothing and preserve long-range information. Extensive experiments on MIMIC-III and MIMIC-IV across multiple clinical prediction tasks demonstrate the effectiveness of `SoAR-SSM`.

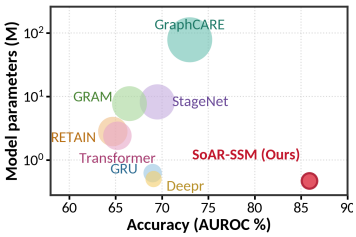

Figure 1: Performance vs model size on MIMIC-IV (mortality).

We summarize the main contributions as follows:

- We introduce a *knowledge-infused temporal hypergraph* representation for patient-specific EHR data, explicitly capturing higher-order clinical interactions within individual visits and their evolution across successive visits.
- We derive a *Sobolev-regularized temporal SSM* that provides a principled topology-dependent resolvent linking hypergraph structure with HiPPO memory. We subsequently generalize its fixed monotone spectral response using parallel ARMA filters, enabling learnable multi-frequency hypergraph representations.
- We provide theoretical guarantees characterizing the stability of the learned representations under perturbations of the hypergraph Laplacian, together with their invariance to permutations of node indices.
- We introduce a temporal reasoning mechanism that extracts attribution based reasoning paths from the learned spatio-temporal representations. These paths are subsequently verbalized using a lightweight language model to generate human-interpretable explanations.

2 PRELIMINARIES

In this section, we introduce the mathematical background required for the proposed framework.

2.1 KNOWLEDGE GRAPHS AND TEMPORAL KNOWLEDGE GRAPHS

Let $\mathcal{V}$ denote a finite set of entities and $\mathcal{R}$ denote a set of relation types. A *knowledge graph* (KG) is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, denotes the set of relational triples. Each triple $e = (s, r, o) \in \mathcal{E}$ represents a relation of type r between a subject s and an object o.

A *temporal knowledge graph* (TKG) extends this representation by associating the relational structure with time. We represent a TKG as a sequence of time-indexed knowledge graphs $\mathcal{T}_{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$, where $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{R}, \mathcal{E}_t)$ denotes the graph observed at time t.

2.2 HYPERGRAPHS

Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ denote a set of N nodes and $\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$ denote a set of M hyperedges, where $e_m \subseteq \mathcal{V}$ and $|e_m| \geq 2$, then a hypergraph is represented as $\mathcal{H} = (\mathcal{V}, \mathcal{E})$. The connectivity between nodes and hyperedges is encoded by the incidence matrix $\mathbf{\Xi}_H \in \mathbb{R}^{N \times M}$. The structure of hypergraph is encoded by hypergraph Laplacian $\bar{\mathbf{L}}_H = \mathbf{I}_N - \mathbf{A}_H$, where $\mathbf{A}_H = \mathbf{D}_v^{-1/2} \mathbf{\Xi}_H \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Xi}_H^\top \mathbf{D}_v^{-1/2}$. Here $\mathbf{D}_v \in \mathbb{R}^{N \times N}$ and $\mathbf{D}_e \in \mathbb{R}^{M \times M}$ denote the node, hyperedge-degree matrices and $\mathbf{W} \in \mathbb{R}^{M \times M}$ denote a diagonal matrix with hyperedge weights. Further we define the normalized shift operator as $\mathbf{L}_H = \left(\frac{\mu_{\max} - \mu_{\min}}{2}\right) \mathbf{I}_N - \bar{\mathbf{L}}_H$, $\mu_{\max}$, $\mu_{\min}$ corresponds to max and min eigenvalues of $\bar{\mathbf{L}}_H$.

3 KNOWLEDGE INFUSED HYPERGRAPH REASONING WITH SSM

In this section, we introduce the proposed `SoAR-SSM` framework. We first present a knowledge infused temporal hypergraph representation. We then introduce a Sobolev regularized dynamic hypergraph-based SSM. Figure 2 illustrates the overall architecture.

3.1 KNOWLEDGE INFUSED HYPERGRAPH REPRESENTATION OF EHR DATA

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_P\}$ denote a set of P patients with each patient P_i being associated with a sequence of clinical visits over time. In particular, considering that patient P_i has T_i visits, indexed by $t \in \{1, 2, \ldots, T_i\}$, clinical data corresponding to patient P_i at visit t is denoted by $C_{i,t} = \{\mathcal{C}_{i,t}, \mathcal{D}_{i,t}, \mathcal{R}_{i,t}\}$, where $\mathcal{C}_{i,t}$, $\mathcal{D}_{i,t}$, and $\mathcal{R}_{i,t}$ represent conditions, drugs, and procedures.

Knowledge Infused Hypergraph Construction: For each visit $t \in T_i$ of patient P_i, we construct a visit-level hypergraph to model higher-order clinical interactions through hyperedges that explicitly captures co-occurring conditions, medications, and procedures within a single visit. A naive approach would invoke a large language model (LLM) to construct a hypergraph independently for each visit. However, such a strategy incurs prohibitive token and computational costs. To address this limitation, we propose a cost-efficient two-stage procedure:

Stage 1: We first construct a *global* hyperknowledge graph (HKG) over the EHR concept vocabulary by aggregating unique clinical concepts across all patients. Using LLM prompting or ontology based clustering, we group semantically related conditions, medications, and procedures into clinically coherent concept sets. Each group defines a hyperedge, capturing meaningful higher-order clinical context. The HKG can be expanded by pooling information from external sources such as PubMed and UMLS along with the derived medical ontologies.

Stage 2: For each visit, we then instantiate a patient-specific hypergraph by selecting the subset of global hyperedges with the concepts in that visit, thereby avoiding repeated LLM calls. To encode visit-level context, we introduce a dedicated *visit node* connected to all hyperedges in the visit level hypergraph. Details on the prompts and hypergraph construction are given in Appendix D.1.

Formally, for a visit t of patient P_i, we define a hypergraph as $\mathcal{H}_{i,t} = (\mathcal{V}_{i,t}, \mathcal{E}_{i,t})$, where $\mathcal{V}_{i,t}$ denotes the set of unique clinical entities appearing in $C_{i,t}$ and visit node, $\mathcal{E}_{i,t}$ is the set of $M_{i,t}$ hyperedges

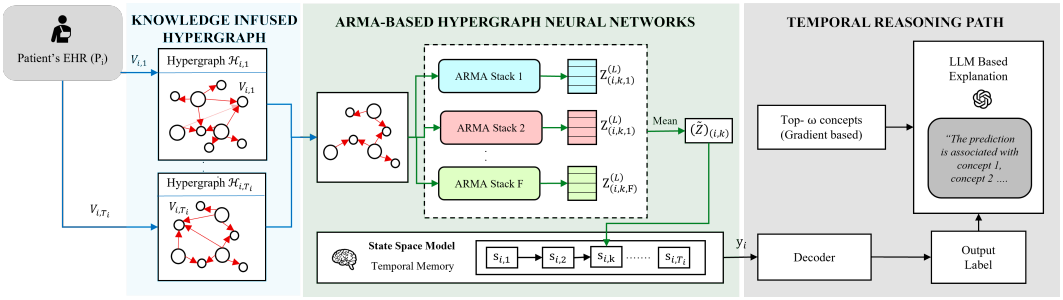

Figure 2: SoAR−SSM: Patient specific temporal hypergraphs, hypergraph based dynamic state space model, reasoning module for natural language explanations.

inferred by the LLM. A single visit can include multiple hyperedges indicating that patient has multiple dissimilar concepts (i.e., $M_{i,t}$). Node features are initialized using BERT embeddings (Devlin et al., 2019) and projected to the latent space via a learnable linear layer.

Figure 3(c) illustrates a knowledge-infused hypergraph constructed from a single visit with clinical concepts shown in Figure 3(a). Figure 3(b) presents the pairwise KG, which models conditions, procedures, and medications independently despite their co-occurrence within the same visit. Consequently, it fails to capture the joint clinical context and interdependencies among neurological, infectious, and respiratory conditions. On the other hand, knowledge-infused hypergraph encodes visit-level information through hyperedges that group clinically related concepts, enabling representation of higher-order disease patterns. For instance, in Figure 3(c), a hyperedge for respiratory disorders jointly connects pleurisy, respiratory intubation, mechanical ventilation, and associated medications. By modeling such multi-way interactions, the hypergraph provides a more expressive representation for richer clinical reasoning and improved downstream performance.

Knowledge Infused Temporal Hypergraphs. For each patient P_i, we construct a sequence of visit-level hypergraphs ordered by time i.e., $\mathcal{H}_i = \{\mathcal{H}_{i,1}, \mathcal{H}_{i,2}, \ldots, \mathcal{H}_{i,T_i}\}$. This enables the proposed model to effectively capture the patient history as temporal hypergraphs. Further, this allows us to apply dynamic representation learning technique to exploit the temporal evolution.

3.2 Sobolev-Regularized Temporal Hypergraph State Space Model

In this section, we derive the hypergraph based temporal state space model from first principles. We first introduce a quadratic regularizer based on the Sobolev operator and incorporate it into the HiPPO approximation objective and derive the temporal state space evolution.

Definition 1 (Sobolev Quadratic Regularizer)**.** (Fischer & Steinwart, 2020) Let $\bar{\mathbf{L}}_H \in \mathbb{R}^{N\times N}$ denote the normalized hypergraph Laplacian, and $\mathbf{X} \in \mathbb{R}^{N\times D}$ denote a D-dimensional signal over the N nodes of the hypergraph. For $\epsilon \geq 0$ and $\beta > 0$, the Sobolev quadratic regularizer is defined as

$$\mathcal{R}_{\beta,\epsilon}(\mathbf{X}) = \frac{1}{2}\,\mathrm{Tr}\left(\mathbf{X}^T\left(\bar{\mathbf{L}}_H + \epsilon\mathbf{I}_N\right)^{\beta}\mathbf{X}\right), \tag{1}$$

where $\mathbf{I}_N$ is an Identity matrix of dimension N and the parameters β and ϵ control the spectral weighting imposed on the hypergraph signal, specifically parameter β allows control over the spectrum. In particular, $\beta = 1$ and $\epsilon = 0$ recover the conventional quadratic smoothness regularizer based on the hypergraph Laplacian. In what follows, we derive the temporal hypergraph SSM for a more general case (not patient specific) and later provide patient specific implementation details.

Sobolev-Regularized HiPPO Memory: Let $\{\mathcal{H}_1, \ldots, \mathcal{H}_T\}$ denote a hypergraph snapshots, where $\bar{\mathbf{L}}_{H,k} \in \mathbb{R}^{N\times N}$ and $\mathbf{X}_k \in \mathbb{R}^{N\times D}$ denote the corresponding hypergraph Laplacian and node-feature matrix at snapshot t_k, respectively. We assume that all snapshots are defined over a common node space of size N, with nodes that are absent from a particular visit appropriately masked. Our objective is to construct a finite-dimensional HiPPO memory that compresses the historical node signals while accounting for the evolving hypergraph structure (Gu et al., 2021; Li et al., 2021).

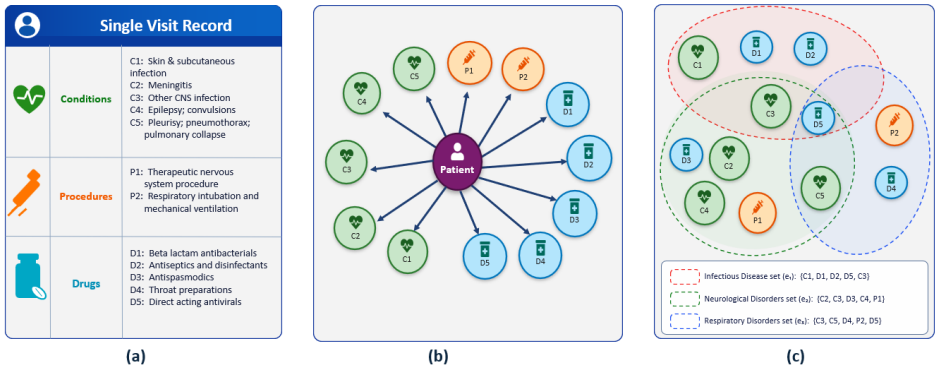

Figure 3: (a) Sample from EHR data. (b) Pairwise knowledge graph. (c) Knowledge hypergraph.

Let $\mathbf{p}(\tau) = [p_0(\tau), \ldots, p_{d-1}(\tau)]^T \in \mathbb{R}^d$ denote d HiPPO basis functions that are orthonormal with respect to the measure $d\mu_k(\tau)$, i.e.,$\int_0^{t_k} \mathbf{p}(\tau)\mathbf{p}(\tau)^T d\mu_k(\tau) = \mathbf{I}_d$. For simplicity, we consider the q-th feature of the hypergraph signal, denoted by $\mathbf{x}_q(\tau) \in \mathbb{R}^N$, and derive a compressed representation of its historical trajectory. For notational convenience, we henceforth index the timestamp t_k simply by k. The trajectory of the q-th feature up to time t_k is approximated using the HiPPO basis as $\mathbf{x}_{q,k}(\tau) = \bar{\mathbf{S}}_{q,k}\mathbf{p}(\tau) + \mathbf{r}_{q,k}(\tau)$, where $\bar{\mathbf{S}}_{q,k} \in \mathbb{R}^{N\times d}$ denotes the HiPPO coefficient matrix and $\mathbf{r}_{q,k}(\tau) \in \mathbb{R}^N$ denotes the residual. To obtain optimal coefficients, while incorporating the structural information of the hypergraph into this temporal compression we solve the following Sobolev-regularized approximation problem:

$$\mathbf{S}_{q,k} = \underset{\bar{\mathbf{S}}_{q,k}\in\mathbb{R}^{N\times d}}{\operatorname{argmin}} \quad \frac{1}{2}\int_0^{t_k} \left\|\mathbf{x}_{q,k}(\tau) - \bar{\mathbf{S}}_{q,k}\mathbf{p}(\tau)\right\|_2^2 d\mu(\tau) \quad + \frac{\lambda}{2}\operatorname{Tr}\left(\bar{\mathbf{S}}_{q,k}^T \left(\bar{\mathbf{L}}_{H,k} + \epsilon\mathbf{I}_N\right)^\beta \bar{\mathbf{S}}_{q,k}\right), \tag{2}$$

where $\lambda > 0$ controls the strength of the Sobolev regularization.

Theorem 3.1 (Sobolev-Regularized HiPPO Coefficients)**.** *Assuming that the HiPPO basis functions satisfy the orthonormality condition. Then, the solution for the regularized HiPPO approximation problem in equation 2 is given by*

$$\mathbf{S}_{q,k} = \underbrace{\left[\mathbf{I}_N + \lambda\left(\bar{\mathbf{L}}_{H,k} + \epsilon\mathbf{I}_N\right)^\beta\right]^{-1}}_{\mathbf{R}_{H,k}} \mathbf{C}_{q,k}, \tag{3}$$

where $\mathbf{C}_{q,k} = \int_0^{t_k} \mathbf{x}_{q,k}(\tau)\mathbf{p}(\tau)^T d\mu_k(\tau) \in \mathbb{R}^{N\times d}$ *denotes the classical HiPPO coefficients.*

Proof of this Theorem is relegated to Appendix B.1. Observe from equation 3 the topology aware memory coefficients are the Sobolev filtered classical HiPPO coefficients. Next, we derive a piecewise linear continuous SSM assuming the filter remains constant between consecutive visits.

Temporal State Space Evolution. Exploiting the connection between the Sobolev-regularized memory and the classical HiPPO coefficients, we derive its temporal evolution. From here on, we drop dependence on feature i.e., q and derive it for general case. Under the standard HiPPO construction, the evolution of $\mathbf{C}$ is given by $\frac{d\mathbf{C}(t)}{dt} = \mathbf{C}(t)\mathbf{A}^T + \mathbf{X}(t)\mathbf{B}^T$, where $\mathbf{A} \in \mathbb{R}^{d\times d}, \mathbf{B} \in \mathbb{R}^{d\times D}$ denotes the HiPPO state matrix, input projection matrix. Since the underlying hypergraph evolves at discrete visits, we consider $\mathbf{L}_{H,k}$ to remain fixed over the interval $(t_{k-1}, t_k]$. Consequently Sobolev filter $\mathbf{R}_{H,k}$ is constant within this interval and piecewise evolution of memory satisfies

$$\begin{aligned}\frac{d\mathbf{S}(t)}{dt} &= \mathbf{R}_{H,k}\frac{d\mathbf{C}(t)}{dt}, \forall t \in (t_{k-1}, t_k] = \mathbf{R}_{H,k}\left(\mathbf{C}(t)\mathbf{A}^T + \mathbf{X}(t)\mathbf{B}^T\right) \\ &= \mathbf{S}(t)\mathbf{A}^T + \mathbf{R}_{H,k}\mathbf{X}(t)\mathbf{B}^T \qquad \forall t \in (t_{k-1}, t_k].\end{aligned} \tag{4}$$

Remark: At a topology change point, the state is interpreted using the operator associated with the new interval. *Importantly, we don't differentiate across such boundary points. Instead at each*

observed topology change, SoAR-SSM follows a diffuse-then-update principle it first diffuses the current visit signal over the newly observed hypergraph and then injects this structural representation into the propagated SSM memory (details in section 3.4).

Discrete-Time State Space Update. For practical implementation, we discretize equation 4 using zero-order hold. Let $\Delta_k = t_k - t_{k-1}$ and assume that $\mathbf{X}(t) = \mathbf{X}_k$ and $\mathbf{L}_H(t) = \mathbf{L}_{H,k}$ for $t \in (t_{k-1}, t_k]$. Then we define $\overline{\mathbf{A}} = \exp\left(\Delta_k \mathbf{A}^T\right) \in \mathbb{R}^{d\times d}$ and $\mathbf{G} = \left(\mathbf{A}^T\right)^{-1}\left[\exp\left(\Delta_k \mathbf{A}^T\right) - \mathbf{I}_d\right]$.

Theorem 3.2 (Discrete-Time Sobolev Hypergraph SSM)**.** *Under the zero-order hold assumption above, the Sobolev-regularized state evolution between two consecutive hypergraph snapshots is*

$$\mathbf{S}_k = \mathbf{S}_{k-1}\overline{\mathbf{A}} + \mathbf{R}_{H,k}\mathbf{X}_k\mathbf{B}^T\mathbf{G}. \tag{5}$$

Proof of this Theorem is relegated to Appendix B.2. Observe that the proposed state space evolution jointly captures long-range temporal dependencies through the HiPPO state dynamics and hypergraph structural information through the Sobolev filtering operator.

3.3 ARMA-Based Hypergraph Neural Networks

The Sobolev filtering operator $\mathbf{R}_{H,k}$ requires careful selection of the spectral parameters, particularly β, to control the relative emphasis across different frequencies and requires a matrix inversion which limits its scalability to large hypergraphs. To address these challenges we adopt a learnable rational filtering approach based on ARMA neural networks. To begin with, we extend ARMA graph neural networks (Bianchi et al., 2021) to hypergraphs while establishing connection to Sobolev response.

Let $\mathbf{X}_k \in \mathbb{R}^{N\times D}$, $\mathbf{L}_{H,k} \in \mathbb{R}^{N\times N}$ denote input node features and the normalized hypergraph shift operator, then ARMA hypergraph neural network for realizing filter of order 1 is given by

$$\mathbf{Z}_f^{(l+1)} = \sigma\left(\mathbf{L}_{H,k}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l)} + \mathbf{X}_k\mathbf{W}_{i,f}^{(l)}\right), \tag{6}$$

where $l = 0, 1, \ldots, L-1$ denotes the recursion depth, $\mathbf{W}_{r,f}^{(l)} \in \mathbb{R}^{d\times d}$, $\mathbf{W}_{i,f}^{(l)} \in \mathbb{R}^{D\times d}$ are the learnable weights associated with the recursive component and input projection. A stack of L recursive updates in equation 6 provides a finite-depth neural realization of an ARMA-1 hypergraph filter . Unlike finite-order polynomial filters, the recursive connection enables an implicit rational spectral response, providing greater flexibility in approximating the Sobolev filter (detailed analysis on the filter response and its connection to Sobolev filter response is given in Appendix C).

For an efficient spectral modeling, ARMA-1 responses are combined. In particular, ARMA-F filter obtains the response by aggregating the output of the F-ARMA-1 filters as

$$\widetilde{\mathbf{Z}} = \frac{1}{F}\sum_{f=1}^{F}\mathbf{Z}_f^{(L)}, \tag{7}$$

where $\mathbf{Z}_f^{(L)} \in \mathbb{R}^{N\times d}$ is the output of the f-th ARMA-1 filter after L recursive updates. The resulting representation $\widetilde{\mathbf{Z}}$ is subsequently used as the Sobolev-filtered hypergraph representation in equation 5.

3.4 Implementation Details

For each patient i, we construct a sequence of visit-level hypergraphs with a visit node following Section 3.1. Then for each hypergraph $\mathcal{H}_{i,k}$, the ARMA hypergraph network in equation 7 produces the spatial representation $\widetilde{\mathbf{Z}}_{i,k} = \frac{1}{F}\sum_{f=1}^{F}\mathbf{Z}_{i,k,f}^{(L)}$, with the visit-level embedding as

$$\widetilde{\mathbf{z}}_{i,k}^{(v)} = (\mathbf{e}_{i,k}^{(v)})^\top\widetilde{\mathbf{Z}}_{i,k} \in \mathbb{R}^{1\times d}, \tag{8}$$

where $\mathbf{e}_{i,k}^{(v)} \in \mathbb{R}^{N_{i,k}\times 1}$ is the canonical vector corresponding to the visit node. The resulting sequence is then propagated through the temporal SSM, where $\mathbf{s}_{i,k} \in \mathbb{R}^{1\times d}$ denotes the patient state after visit k. While the derivation considers the memory feature-wise, the ARMA encoder jointly projects node features into a shared d-dimensional latent space for parameter efficiency. *Importantly, rather*

than imposing persistent node-wise temporal states across visits with varying clinical entity sets, SoAR-SSM performs topology-aware higher-order diffusion within each visit and propagates the resulting visit-node structural readout through the HiPPO SSM, thereby explicitly separating intra-visit hypergraph reasoning from inter-visit long-range dynamics. With $\mathbf{s}_{i,0} = \mathbf{0}$ state evolves as

$$\mathbf{s}_{i,k} = \mathbf{s}_{i,k-1}\overline{\mathbf{A}} + \widetilde{\mathbf{z}}_{i,k}^{(v)}\mathbf{B}^T\mathbf{G}, \tag{9}$$

here $\mathbf{B} \in \mathbb{R}^{d\times d}$. After processing all T_i visits, the final patient representation is obtained as $\mathbf{y}_i = \mathbf{s}_{i,T_i}\mathbf{C}^T \in \mathbb{R}^{1\times d_o}$, where $\mathbf{C} \in \mathbb{R}^{d_o\times d}$ is a learnable output projection matrix. The resulting representation is subsequently passed to a task-specific decoder for downstream clinical prediction. Further leveraging the representations from SSM we adopt an gradient based approach (details in Appendix D.2) and extract the temporal reasoning paths responsible for models prediction. These are subsequently fed to a light weight LLM for a natural language and more interpretable explanations.

4 Theoretical Guarantees

Theorem 4.1. *(Robustness to Hypergraph Perturbations) Let $\widehat{\mathbf{L}}_{H,k} = \mathbf{L}_{H,k} + \Delta\mathbf{L}_{H,k}$ be the perturbed shift operator with $\|\Delta\mathbf{L}_{H,k}\|_2 \le \varepsilon$. Under the following assumptions (a) $\sigma(\cdot)$ being C-Lipschitz continuous. (b) $\|\mathbf{W}_{r,f}^{(l)}\|_2 \le \Theta, \|\mathbf{W}_{i,f}^{(l)}\|_2 \le \Omega$ for every ARMA filter (c) $\|\hat{\mathbf{L}}_{H,k}\|_2 \le \rho$ (d) $\mathbf{Z}_f^0 = \mathbf{0}$ the error between the spatial representations is bounded as $\left\|\widehat{\widetilde{\mathbf{Z}}} - \widetilde{\mathbf{Z}}\right\|_F \le \mathrm{const}\varepsilon\|\mathbf{X}_k\|_F$, where* $\mathrm{const} = C\kappa\Theta\sum_{l=0}^{L-1}(C\rho\Theta)^l$, *with* $\kappa = C\Omega\sum_{j=0}^{L-1}(C\rho\Theta)^j$. *Furthermore, if $\|\mathbf{B}\|_2 \le \gamma$ and $\|\mathbf{G}\|_2 \le \delta$, then with identical initialization, the patient state satisfies*

$$\|\widehat{\mathbf{s}}_{i,k} - \mathbf{s}_{i,k}\|_2 \le \gamma\delta\mathrm{const}\varepsilon\|\mathbf{X}_{i,k}\|_F. \tag{10}$$

The proof for this Theorem is relegated to Appendix B.3. It shows that under the perturbation of the hypergraphs the representations from SoAR-SSM scales linearly with perturbed matrix energy.

Corollary 1 (Error Accounting for Temporal Propagation)**.** Assume that the discretized state transition matrix satisfies $\|\overline{\mathbf{A}}\|_2 \le \eta$ for all k, where $\overline{\mathbf{A}} = \exp(\Delta_k\mathbf{A}^T)$. Then, after K temporal updates, the error between the perturbed and unperturbed patient representations satisfies $\|\widehat{\mathbf{s}}_{i,K} - \mathbf{s}_{i,K}\|_2 \le \mathrm{const}_1\,\epsilon$, where $\mathrm{const}_1 = \gamma\delta\mathrm{const}\sum_{\tau=1}^{K}\eta^{K-\tau}\|\mathbf{X}_{i,\tau}\|_F$,

The proof for this corollary is relegated to B.4. It can be observed even while accounting for the temporal propagation, the error scales linearly with the energy of the perturbation matrix.

Theorem 4.2 (Permutation Invariance)**.** *Let $\mathcal{P} = \{\mathbf{P} \in \{0,1\}^{N\times N} : \mathbf{P}^\top\mathbf{P} = \mathbf{P}\mathbf{P}^\top = \mathbf{I}_N\}$ be the set of all valid permutation matrices . Under the permutation of node indices of the visit-level hypergraph, the representation from the hypergraph ARMA neural network remains equivariant, i.e., $\tilde{\mathbf{Z}}^{\mathrm{perm}} = \mathbf{P}\tilde{\mathbf{Z}}$. However, the output of the dynamic hypergraph SSM module remains invariant.*

The proof of this theorem is relegated to Appendix B.5. It states that if node labels are permuted then only the ARMA representations are relabeled whereas SSM output remains unchanged.

5 Numerical Experiments

In this section, we present results on the MIMIC III (Johnson et al., 2016) and MIMIC IV (Johnson et al., 2021) data with downstream tasks as mortality, length of stay (LOS), drug recommendation, phenotyping and diagnosis prediction task. More details on datasets and tasks in Appendix E.

Implementation details. Visit-level hypergraphs for each patient are generated using Azure GPT-4.1 as described in Section 3.1 with prompts as in Fig. 6 and Fig 7. Then spatial representations are computed with proposed hypergraph ARMA model and its counterpart based on convolution operator (that typically models data in the low frequency range). Accordingly, we consider two variants of the proposed framework: SoAR-SSM (v1), which employs hypergraph convolution as the spatial encoder, and SoAR-SSM, which employs ARMA model. The resulting spatial representations are propagated over time using the hypergraph-based state-space model (SSM) to produce the final predictions. Whereas the temporal reasoning paths are generated following the procedure in Sec D.2. More details on hyperparameters and training configuration in Appendix E.4.

Table 1: Performance comparison on mortality prediction and drug recommendation using MIMIC-III and MIMIC-IV. Results are reported as mean$_{(\text{std})}$. Best results are highlighted in blue and bold.

Model	Mortality Prediction				Drug Recommendation			
	MIMIC-III		MIMIC-IV		MIMIC-III		MIMIC-IV	
	AUPRC	AUROC	AUPRC	AUROC	Jaccard	F1	Jaccard	F1
GRU	$11.8_{(0.5)}$	$61.3_{(0.9)}$	$4.2_{(0.1)}$	$69.0_{(0.8)}$	$47.8_{(0.3)}$	$62.3_{(0.3)}$	$44.0_{(0.4)}$	$60.2_{(0.2)}$
Transformer	$10.1_{(0.9)}$	$57.2_{(1.3)}$	$4.0_{(0.4)}$	$65.1_{(1.2)}$	$47.1_{(0.4)}$	$62.1_{(0.4)}$	$40.4_{(0.1)}$	$55.9_{(0.2)}$
RETAIN	$9.6_{(0.6)}$	$59.4_{(1.5)}$	$3.8_{(0.4)}$	$64.8_{(1.6)}$	$48.8_{(0.2)}$	$63.7_{(0.2)}$	$45.0_{(0.1)}$	$60.3_{(0.1)}$
GRAM	$11.4_{(0.7)}$	$60.4_{(0.9)}$	$4.4_{(0.3)}$	$66.7_{(0.7)}$	$47.9_{(0.3)}$	$62.9_{(0.3)}$	$45.3_{(0.3)}$	$60.1_{(0.1)}$
Deepr	$13.2_{(1.1)}$	$60.8_{(0.4)}$	$4.2_{(0.2)}$	$68.9_{(0.9)}$	$44.7_{(0.3)}$	$60.3_{(0.4)}$	$43.8_{(0.4)}$	$59.1_{(0.4)}$
StageNet	$12.4_{(0.3)}$	$61.5_{(0.7)}$	$4.2_{(0.3)}$	$69.6_{(0.8)}$	$45.8_{(0.4)}$	$61.4_{(0.3)}$	$45.4_{(0.4)}$	$60.2_{(0.3)}$
GraphCARE	$16.7_{(0.5)}$	$70.3_{(0.5)}$	$6.7_{(0.3)}$	$73.1_{(0.5)}$	$49.8_{(0.4)}$	$66.8_{(0.2)}$	$48.1_{(0.3)}$	$63.9_{(0.3)}$
SoAR-SSM (v1)	$34.6_{(1.0)}$	$73.7_{(0.5)}$	$23.8_{(0.9)}$	$84.7_{(0.2)}$	$52.2_{(0.5)}$	$66.5_{(0.4)}$	$50.6_{(0.6)}$	$64.1_{(0.8)}$
SoAR-SSM	**$36.1_{(1.3)}$**	**$76.2_{(1.0)}$**	**$26.9_{(0.6)}$**	**$85.9_{(0.1)}$**	**$53.2_{(0.5)}$**	**$67.9_{(0.9)}$**	**$54.3_{(0.4)}$**	**$68.3_{(0.4)}$**

5.1 Discussion on Results

In Tables 1 and 2 we present results on the MIMIC-III and MIMIC-IV datasets. It can be observed that SoAR-SSM demonstrates strong performance across all tasks compared to the baselines. Importantly, on mortality prediction, it achieves a improvement of **+19.4% in AUPRC** and **+5.9% in AUROC** over GraphCARE on the MIMIC-III dataset, asserting the importance of modeling higher-order relations, information across different frequency regimes and long range information for critical tasks. Whereas, on MIMIC-IV, we observe that the relative gains are stronger with **+20.2% in AUPRC** and **+12.8% in AUROC** improvements, as the dataset includes patients with more visits, which showcases the ability of the model to capture long-range patient history. Importantly, compared to SoAR-SSM (v1) that typically models data on low-frequency regime SoAR-SSM outperforms asserting importance of modeling data from diverse spectrum for critical tasks. Additional results on phenotyping task and diagnosis prediction task are provided in Appendix F.3. Reasoning paths together with an interpretable explanation by LLM is presented in Fig. 4.

5.2 Ablation Study 1: Effect of Hypergraph Modeling

To study the impact of modeling higher-order clinical relationships with hypergraphs, we compare SoAR-SSM against its graph-based counterpart GNN-SSM that models relationships using knowledge graphs, similar to GraphCare (Jiang et al., 2024). Table 3 reports the results on the mortality prediction task for both the MIMIC-III and MIMIC-IV datasets. As shown, SoAR-SSM consistently outperforms GNN-SSM and graph ARMA based model (ARMA-SSM), where GNN is replaced by graph ARMA network demonstrating the advantage of hypergraph modeling.

5.3 Ablation study 2: Effect of using SSM

To examine the impact of long-range temporal information propagation on performance, we conduct an ablation study on a subset of patients with histories (more than 10 recorded visits) on mortality prediction task on the MIMIC-IV dataset. This setting is particularly challenging, as accurate prediction requires preserving clinically relevant information over long temporal horizons. We compare SoAR-SSM against two architectural variants with SSM module is replaced by (i) Long Short-Term Memory (LSTM), (SoAR-LSTM), and (ii) Hyper-GNN, which directly aggregates visit-level hypergraph representations without temporal modelling. In Table 4, we report the results where it can be observed that SoAR-SSM consistently outperforms both variants across all evaluation metrics. Notably, SoAR-SSM achieves larger gains with **+17% in Sensitivity** and **+7.2% in Macro-F1 score** compared to the LSTM- baseline, asserting the importance of SoAR-SSM in capturing the long range clinical histories especially in critical tasks such as mortality prediction. More importantly, one can observe the importance of preserving the sequential (temporal) information compared to static hypergraph based approaches (Hyper-GNN) on critical tasks.

Table 2: Performance on LOS prediction task.

Model	MIMIC-III		MIMIC-IV	
	Kappa	F1	Kappa	F1
GRU	$26.2_{(0.2)}$	$34.9_{(0.5)}$	$26.0_{(0.1)}$	$31.6_{(0.2)}$
Transformer	$25.4_{(0.4)}$	$34.8_{(0.2)}$	$25.3_{(0.4)}$	$31.4_{(0.3)}$
RETAIN	$26.1_{(0.4)}$	$34.9_{(0.4)}$	$26.3_{(0.2)}$	$32.0_{(0.2)}$
GRAM	$26.3_{(0.3)}$	$34.5_{(0.2)}$	$26.1_{(0.4)}$	$31.9_{(0.3)}$
Deepr	$25.3_{(0.4)}$	$35.0_{(0.4)}$	$26.4_{(0.2)}$	$32.3_{(0.1)}$
StageNet	$24.8_{(0.3)}$	$34.4_{(0.4)}$	$26.0_{(0.2)}$	$31.3_{(0.3)}$
GraphCARE	$29.5_{(0.4)}$	$37.5_{(0.2)}$	$29.8_{(0.3)}$	$34.2_{(0.3)}$
SoAR-SSM$_{(v1)}$	$30.2_{(0.8)}$	$37.7_{(0.8)}$	$27.5_{(0.3)}$	$34.0_{(0.3)}$
SoAR-SSM	**31.3**$_{(0.7)}$	**39.4**$_{(0.6)}$	**30.5**$_{(0.4)}$	**35.8**$_{(0.2)}$

Table 3: Ablation on hypergraphs.

Model	MIMIC-III		MIMIC-IV	
	AUPRC	AUROC	AUPRC	AUROC
GNN-SSM	22.9	69.5	19.8	77.4
ARMA-SSM	24.7	71.3	20.6	79.7
SoAR-SSM	**36.1**	**76.2**	**26.9**	**85.9**

Table 4: Ablation on long-range task.

Model	AUROC	Macro F1	Sensitivity
Hyper-GNN	60.6	62.8	25.0
SoAR-LSTM	65.1	68.9	40.0
SoAR-SSM	**69.3**	**76.1**	**57.0**

Mortality label=1, Predicted label=1
Visit 1:
↪ **Entities:** *cardiac arrest and ventricular fibrillation, septicemia*
↓
Visit 2:
↪ **Entities:** *other liver diseases, ace inhibitors, plain*
↓
Visit 3:
↪ **Entities:** *chronic ulcer of skin, ace inhibitors, plain*
LLM explanation: This patient's clinical trajectory reveals a progressively deteriorating health status. Initial visits show life-threatening conditions such as cardiac arrest and ventricular fibrillation, alongside septicemia, which are indicative of a severe systemic infection. Subsequent visits reveal the presence of other liver diseases, chronic skin ulcers, and use of ACE inhibitors, suggesting underlying organ dysfunction and compromised wound healing. The persistence of these serious conditions over multiple visits suggests a complex, multi-systemic disease process that is likely to lead to in-hospital mortality.

Figure 4: Gradient based temporal reasoning path.

Further, to ensure that the observed gains are not solely due to LLM-based hyperedge construction, to understand different spectral responses encoded by hypergraph ARMA we investigate the following questions (details in Appendix F.1, D.5).

- **Q1**. How does SoAR-SSM perform with non-LLM based hypergraph constructions?
- **Q2**. How robust and reliable are the performance gains?
- **Q3**. How effectively does SoAR-SSM capture different spectral responses and avoid oversmoothing?

6 Conclusions

We proposed SoAR-SSM, a novel higher-order temporal reasoning framework that models higher order interactions, multi-scale frequency information and long range information. The proposed framework constructs knowledge infused temporal hypergraphs to encode co-occurring clinical concepts within visits and proposes an ARMA hypergraph SSM to capture long-range dependencies across patient visits. We also extract reasoning paths based on gradient approaches, allowing LLM-assisted explanation for the interpretable inference. Results on MIMIC-III and MIMIC-IV demonstrate that SoAR-SSM effectively captures higher-order relations, multi-scale information and long-range temporal information, leading to substantial gains in the predictive performance.

AI USE STATEMENT

We used LLM tools to assist with grammar, rephrasing, and language polishing of author-drafted text. We thoroughly reviewed them again. Starting from preprocessing of the data to development of source code for the method is completely written by authors. The authors take full responsibility for the final content of this work, including its text, figures, claims, experimental methodology, and results.

ETHICS STATEMENT

This work uses only publicly available datasets and models and does not involve human subjects, personal data, or sensitive attributes. Wherever appropriate the datasets have been properly cited and adhered to applicable data and model licensing terms and to the ICLR Code of Ethics. As the model developed involves the experimentation on crucial tasks such as mortality prediction we encourage responsible deployment with appropriate safeguards and human oversight.

REPRODUCIBILITY STATEMENT

We have included the code as supplementary material to reproduce all the results presented in the paper. Details such as training configurations, prompts, hyperparameters of the model and the data preprocessing strategy are discussed clearly in the Appendix.

REFERENCES

Bader Aldughayfiq, Farzeen Ashfaq, NZ Jhanjhi, and Mamoona Humayun. Capturing semantic relationships in electronic health records using knowledge graphs: An implementation using mimic iii dataset and graphdb. In *Healthcare*, volume 11, pp. 1762. MDPI, 2023.

Negin Ashrafi, Armin Abdollahi, Jiahong Zhang, and Maryam Pishgar. Optimizing mortality prediction for icu heart failure patients: Leveraging xgboost and advanced machine learning with the mimic-iii database. *arXiv preprint arXiv:2409.01685*, 2024.

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3496–3507, 2021.

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

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016b.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 787–795, 2017.

Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 606–613, 2020.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

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

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *Journal of Machine Learning Research*, 21(205):1–38, 2020.

Junyi Gao, Cao Xiao, Yasha Wang, Wen Tang, Lucas M Glass, and Jimeng Sun. Stagenet: Stage-aware neural networks for health risk prediction. In *Proceedings of the web conference 2020*, pp. 530–540, 2020.

Yanjun Gao, Ruizhe Li, Emma Croxford, John Caskey, Brian W Patterson, Matthew Churpek, Timothy Miller, Dmitriy Dligach, and Majid Afshar. Leveraging medical knowledge graphs into large language models for diagnosis prediction: Design and application study. *Jmir Ai*, 4:e58670, 2025.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Amelie Gyrard, Manas Gaur, Saeedeh Shekarpour, Krishnaprasad Thirunarayan, and Amit Sheth. Personalized health knowledge graph. In *CEUR workshop proceedings*, volume 2317, pp. 5, 2018.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

Pengcheng Jiang, Cao Xiao, Adam Richard Cross, and Jimeng Sun. Graphcare: Enhancing healthcare predictions with personalized knowledge graphs. In *The Twelfth International Conference on Learning Representations*, 2024.

Pengcheng Jiang, Cao Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xinke Jiang, Ruizhe Zhang, Yongxin Xu, Rihong Qiu, Yue Fang, Zhiyuan Wang, Jinyi Tang, Hongxin Ding, Xu Chu, Junfeng Zhao, et al. Think and retrieval: A hypothesis knowledge graph enhanced medical large language models. *CoRR*, 2023.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and S Horng IV. Mimic-iv-ed. *PhysioNet*, 2021.

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. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Mohsen Nayebi Kerdabadi, Arya Hadizadeh Moghaddam, Chen Chen, Dongjie Wang, and Zijun Yao. Text-attributed knowledge graph enrichment with large language models for medical concept representation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16544–16560, 2026a.

Mohsen Nayebi Kerdabadi, Arya Hadizadeh Moghaddam, Chen Chen, Dongjie Wang, and Zijun Yao. Text-attributed knowledge graph enrichment with large language models for medical concept representation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16544–16560, 2026b.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *Advances in Neural Information Processing Systems*, 37:127030–127058, 2024.

Yang Li, Buyue Qian, Xianli Zhang, and Hui Liu. Graph neural network-based diagnosis prediction. *Big data*, 8(5):379–390, 2020.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 408–417, 2021.

Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 743–752, 2018.

Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 833–840, 2020.

Christopher J McWilliams, Daniel J Lawson, Raul Santos-Rodriguez, Iain D Gilchrist, Alan Champneys, Timothy H Gould, Mathew JC Thomas, and Christopher P Bourdeaux. Towards a decision support tool for intensive care discharge: machine learning algorithm development using electronic healthcare data from mimic-iii and bristol, uk. *BMJ open*, 9(3):e025925, 2019.

Mohsen Nayebi Kerdabadi, Arya Hadizadeh Moghaddam, Dongjie Wang, and Zijun Yao. Multi-ontology integration with dual-axis propagation for medical concept representation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 2201–2211, 2025a.

Mohsen Nayebi Kerdabadi, Arya Hadizadeh Moghaddam, Dongjie Wang, and Zijun Yao. Multi-ontology integration with dual-axis propagation for medical concept representation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 2201–2211, 2025b.

Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. `Deepr`: A convolutional net for medical records. *IEEE Journal of Biomedical and Health Informatics*, 21(1):22–30, 2017. doi: 10.1109/JBHI.2016.2633963.

Peipei Ping, Karol Watson, Jiawei Han, and Alex Bui. Individualized knowledge graph: a viable informatics path to precision medicine. *Circulation research*, 120(7):1078–1080, 2017.

Sola Shirai, Oshani Seneviratne, and Deborah L McGuinness. Applying personal knowledge graphs to health. *arXiv preprint arXiv:2104.07587*, 2021.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

Chenhao Su, Sheng Gao, and Si Li. Gate: graph-attention augmented temporal neural network for medication recommendation. *IEEE Access*, 8:125447–125458, 2020.

Hao Tang, Ling Shao, Zhenyu Zhang, Luc Van Gool, and Nicu Sebe. Spatial-temporal graph mamba for music-guided dance video synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 47(11):9626–9636, 2025.

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

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W White, Longqi Yang, et al. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 5836–5847, 2024.

Jenna Wong, Mara Murray Horwitz, Li Zhou, and Sengwee Toh. Using machine learning to identify health outcomes from electronic health record data. *Current epidemiology reports*, 5(4):331–342, 2018.

Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187*, 2024.

Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. Medical graph rag: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28443–28467, 2025.

Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 649–658, 2019.

Yuzhang Xie, Xu Han, Ran Xu, Xiao Hu, Jiaying Lu, and Carl Yang. Hypkg: Hypergraph-based knowledge graph contextualization for precision healthcare. In *International Semantic Web Conference*, pp. 328–348. Springer, 2025.

Ran Xu, Mohammed K Ali, Joyce C Ho, and Carl Yang. Hypergraph transformers for ehr-based clinical predictions. *AMIA Summits on Translational Science Proceedings*, 2023:582, 2023.

Nianzu Yang, Kaipeng Zeng, Qitian Wu, and Junchi Yan. Molerec: Combinatorial drug recommendation with substructure-aware molecular representation learning. In *Proceedings of the ACM web conference 2023*, pp. 4075–4085, 2023a.

Zhichao Yang, Avijit Mitra, Weisong Liu, Dan Berlowitz, and Hong Yu. Transformehr: transformer-based encoder-decoder generative model to enhance prediction of disease outcomes using electronic health records. *Nature communications*, 14(1):7857, 2023b.

Chaohe Zhang, Xin Gao, Liantao Ma, Yasha Wang, Jiangtao Wang, and Wen Tang. Grasp: generic framework for health status representation learning based on incorporating knowledge from similar patients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 715–723, 2021.

Weicheng Zhu and Narges Razavian. Variationally regularized graph-based representation learning for electronic health records. In *Proceedings of the conference on health, inference, and learning*, pp. 1–13, 2021.

APPENDIX

1. Prior Works 14
2. Proof of Theorems 15
3. ARMA recursive realization 19
4. Implementation Details 19
5. Dataset and Training Details 25
6. Additional Experiments 27
7. Reasoning Path Examples 30
8. Future Directions 30

A Prior Works

Prior research on electronic health record (EHR) data for downstream clinical prediction and recommendation tasks can be broadly categorized into representation learning-based approaches and methods centered on KGs and LLMs.

Representation Learning–Based Approaches. Early work in this category applied classical machine learning models and deep learning architectures to learn low-dimensional representations of patient data, which were subsequently used for downstream clinical tasks (Wong et al., 2018; McWilliams et al., 2019; Choi et al., 2016a;b; Zhang et al., 2021; Gao et al., 2020; Ashrafi et al., 2024; Chung et al., 2014). While effective, these methods largely overlook the relational graph structure underlying the data.

To explicitly model such relationships, graph neural network (GNN)-based approaches have been proposed, where clinical entities are represented as nodes and their interactions as edges (Choi et al., 2020; Su et al., 2020; Li et al., 2020; Zhu & Razavian, 2021; Yang et al., 2023a; Xie et al., 2019). These models construct patient-specific or visit-level graphs from EHR data and leverage message passing to learn relational representations. However, most representation learning-based approaches operate on local graphs derived solely from EHR records and do not incorporate external medical knowledge sources, such as PubMed or UMLS, which contain rich and complementary relational information. Importantly, learning based approaches are built on using message passing or finite impulse response filters to obtain representations, they typically model the information in the low frequency regime. On the other hand, some works leverage foundation models like transformers (TransformEHR) (Yang et al., 2023b), state space models (EHRMamba) (Fallahpour et al., 2024) for representation learning . For instance, EHRMamba is an adaptation of Mamba-based state space models to EHR data, where clinical events are flattened into token sequences and processed using a sequence model. While this enables efficient long-range dependency modelling, such flattening inherently treats clinical concepts as independent tokens and relies on the model to implicitly learn relationships. As a result, it often fails to explicitly capture co-occurrence patterns and higher-order clinical interactions within a visit.

A separate line of work, largely independent of healthcare applications, extends state space models (SSMs) to structured and temporal data (Gu et al., 2020; Li et al., 2024; Wan et al., 2024; Tang et al., 2025). In particular, Li et al. (2024) develops SSMs for dynamic graphs through a smoothness-based formulation that predominantly captures low-frequency graph structure. In contrast, we introduce an SSM formulation for dynamic hypergraphs based on *Sobolev smoothness*, enabling the model to capture a broader spectrum of both low- and high-frequency structural variations. To the best of our knowledge, this is the first work to formulate hypergraph SSMs through Sobolev smoothness and to establish an explicit connection between the resulting Sobolev filters and ARMA graph filters. Beyond this architectural contribution, we provide theoretical guarantees characterizing the stability of the learned representations under structural perturbations, together with permutation invariance of the proposed hypergraph representation.

Approaches centered on Knowledge Graphs and LLMs Personalized medical knowledge graphs (MKGs) provide a structured representation of EHR data by explicitly modeling semantic relationships among clinical entities. Early approaches in this direction constructed MKGs using predefined hierarchical or rule-based schemas and demonstrated notable performance improvements on downstream clinical tasks (Ping et al., 2017; Gyrard et al., 2018; Shirai et al., 2021). Despite their effectiveness, these methods are limited in their ability to capture complex and higher-order clinical interactions.

With recent advances in large language models (LLMs) pretrained on extensive medical corpora, more recent work has explored the construction of personalized knowledge graphs using LLMs, often augmented with external biomedical resources such as UMLS and PubMed (Jiang et al., 2024; 2025; Gao et al., 2025; Nayebi Kerdabadi et al., 2025b; Kerdabadi et al., 2026a). While these approaches enrich the semantic coverage of MKGs, they typically restrict interactions to pairwise relations. As a result, they fail to capture co-occurring clinical concepts within a single visit and lack explicit mechanisms for modeling temporal evolution or preserving long-range clinical dependencies across a patient's history. Importantly, the representations learnt using message passing approaches are prone to oversmoothing.

Recent works leverage hypergraphs (Xu et al., 2023; Xie et al., 2025) to model EHR data. However they model each visit as a single hyperedge and multiple visits through static hypergraph, thereby failing to explicitly capture intra-visit structure (whose importance is discussed in the Introduction and Section 3.1) as well as inter-visit temporal dynamics. This limitation restricts their ability to effectively model disease progression. More importantly, these approaches lack an explicit mechanism to capture long-range patient information over time. Also these approaches lack the mechanism to model the data at different frequency regimes.

B Proof of Theorems

B.1 Proof for Theorem 3.1

Proof. Recall the minimization problem

$$\mathbf{S}_{q,k} = \underset{\bar{\mathbf{S}}_{q,k}\in\mathbb{R}^{N\times d}}{\operatorname{argmin}} \quad \frac{1}{2}\int_0^{t_k} \left\|\mathbf{x}_{q,k}(\tau) - \bar{\mathbf{S}}_{q,k}\mathbf{p}(\tau)\right\|_2^2 d\mu(\tau) \quad + \frac{\lambda}{2}\operatorname{Tr}\left(\bar{\mathbf{S}}_{q,k}^T\left(\bar{\mathbf{L}}_{H,k}+\epsilon\mathbf{I}_N\right)^{\beta}\bar{\mathbf{S}}_{q,k}\right), \tag{11}$$

where $\lambda > 0$ controls the strength of the Sobolev regularization.

An optimal solution is obtained by using the first order optimality condition $\nabla_{\bar{\mathbf{S}}_{q,k}} = \mathbf{0}$. In particular differentiating equation 11 obtains

$$\begin{aligned}\int_0^{t_k} -(\mathbf{x}_{q,k}(\tau) - \mathbf{S}_{q,k}\mathbf{p}(\tau))\mathbf{p}(\tau)^T d\mu(\tau) + \lambda(\bar{\mathbf{L}}_{H,k}+\epsilon\mathbf{I}_N)^{\beta}\mathbf{S}_{q,k} = \mathbf{0}\\ \mathbf{S}_{q,k} - \mathbf{C}_{q,k} + \lambda\left(\bar{\mathbf{L}}_{H,k}+\epsilon\mathbf{I}_N\right)^{\beta}\mathbf{S}_{q,k} \overset{(a)}{=} \mathbf{0},\end{aligned} \tag{12}$$

To arrive at equation 12(a), we have used $\int_0^{t_k}\mathbf{p}(\tau)\mathbf{p}(\tau)^T d\mu(\tau) = \mathbf{I}_d$ and $\int_0^{t_k}\mathbf{x}_{q,k}\mathbf{p}(\tau)^T d\mu(\tau) = \mathbf{C}_{q,k}$, where $\mathbf{C}_{q,k}$ are the general HiPPO coefficients. Then the solution for Sobolev regularized HiPPO are obtained by

$$\mathbf{S}_{q,k} = (\mathbf{I}_N + \lambda\left(\bar{\mathbf{L}}_{H,k}+\epsilon\mathbf{I}_N\right)^{\beta})^{-1}\mathbf{C}_{q,k} \tag{13}$$

For convenience, we define the Sobolev filtering operator associated with the hypergraph at t_k instant as $\mathbf{R}_{H,k} = \left[\mathbf{I}_N + \lambda\left(\bar{\mathbf{L}}_{H,k}+\epsilon\mathbf{I}_N\right)^{\beta}\right]^{-1}$, then equation 13 can be compactly written as $\mathbf{S}_{q,k} = \mathbf{R}_{H,k}\mathbf{C}_{q,k}$. □

B.2 Proof for Theorem 3.2

Proof. The continous time state space model can be compactly written as

$$\frac{d\mathbf{S}}{dt} = \mathbf{S}\mathbf{A}^T + \mathbf{R}_{H,k}\mathbf{X}_k\mathbf{B}^T. \tag{14}$$

The solution for equation 14 is given as

$$\mathbf{S}(t) = \mathbf{S}(t_{k-1})\exp((\mathrm{t}-\mathrm{t_{k-1}})\mathbf{A}^{\mathrm{T}}) + \int_{\mathrm{t_{k-1}}}^{\mathrm{t}} \mathbf{R}_{\mathrm{H,k}}\mathbf{X}_{\mathrm{k}}\mathbf{B}^{\mathrm{T}}\exp((\mathrm{t}-\tau)\mathbf{A}^{\mathrm{T}})\mathrm{d}\tau$$

Evaluating at t_k and also by realizing that $\Delta_k = t_k - t_{k-1}$ we have

$$\mathbf{S}_k = \mathbf{S}_{k-1}\exp(\Delta_{\mathrm{k}}\mathbf{A}^{\mathrm{T}}) + \int_{\mathrm{t_{k-1}}}^{\mathrm{t_k}} \mathbf{R}_{\mathrm{H,k}}\mathbf{X}_{\mathrm{k}}\mathbf{B}^{\mathrm{T}}\exp((\mathrm{t}-\tau)\mathbf{A}^{\mathrm{T}})\mathrm{d}\tau \tag{15}$$

with change of variable as $t_k - \tau = p$, we further simplify second term in equation 15 as

$$\begin{aligned}\mathbf{G} &= \int_0^{\Delta_k} \exp(\mathrm{p}\mathbf{A}^{\mathrm{T}})\mathrm{dp} \\ &= (\mathbf{A}^T)^{-1}(\exp(\Delta_{\mathrm{k}}\mathbf{A}^{\mathrm{T}} - \mathbf{I}_{\mathrm{d}}))\end{aligned} \tag{16}$$

On using equation 16 in equation 15 and by considering $\bar{\mathbf{A}} = \exp(\Delta_{\mathrm{k}}\mathbf{A}^{\mathrm{T}})$, we have

$$\mathbf{S}_k = \mathbf{S}_{k-1}\bar{\mathbf{A}} + \mathbf{R}_{H,k}\mathbf{X}_k\mathbf{B}^T\mathbf{G}. \tag{17}$$

□

B.3 Proof for Theorem 4.1

Proof. To begin with, we first present the proof for bounding the error between the spatial representations obtained through ARMA hypergraph neural networks. Then leveraging this, we present the proof for bounding the error between the state space representations. Also with slight abuse of notation we represent $\mathbf{L}_{H,k}$ as $\mathbf{L}$ and $\mathbf{X}_k$ as $\mathbf{X}_k$.

Recall, the ARMA-1 obtains the representations following

$$\mathbf{Z}_f = \sigma\left(\mathbf{L}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,F}^{(l)} + \mathbf{X}\mathbf{W}_{i,f}^{(l)}\right). \tag{18}$$

Under the perturbation of the hypergraph structure, the error between the representations are given as

$$\begin{aligned}\left\|\hat{\mathbf{Z}}_f^{(l+1)} - \mathbf{Z}_f^{(l+1)}\right\|_F &= \left\|\sigma\left(\hat{\mathbf{L}}\hat{\mathbf{Z}}_f^{(l)}\mathbf{W}_{r,F}^{(l)} + \mathbf{X}\mathbf{W}_{i,f}^{(l)}\right) - \sigma\left(\mathbf{L}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,F}^{(l)} + \mathbf{X}\mathbf{W}_{i,f}^{(l)}\right)\right\|_F, \\ &\overset{(a)}{\leq} C\left\|\hat{\mathbf{L}}\hat{\mathbf{Z}}_f^{(l)}\mathbf{W}_{r,f}^{(l)} - \mathbf{L}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l)}\right\|_F, \\ &\overset{(b)}{\leq} C\left\|(\hat{\mathbf{L}}-\mathbf{L})\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l)}\right\|_F + \left\|\hat{\mathbf{L}}(\hat{\mathbf{Z}}_f^{(l)} - \mathbf{Z}_f^{(l)})\mathbf{W}_{r,f}^{(l)}\right\|_F, \\ &\overset{(d)}{\leq} C\kappa\epsilon\Theta\|\mathbf{X}\|_F + \left\|(\hat{\mathbf{Z}}_f^{(l)} - \mathbf{Z}_f^{(l)})\right\|_F C\Theta\rho,\end{aligned} \tag{19}$$

where equation 19(a) follows from Lipschitz continuity of $\sigma(.)$ and equation 19(b) follows from rearranging terms by addition and subtraction of $\hat{\mathbf{L}}_H\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l)}$. Whereas equation 19(c) follows since $\|\hat{\mathbf{L}}\|_2 \leq \rho$ and equation 19(d) follows by leveraging the ARMA recursion and the above assumptions,

$$\|\mathbf{Z}_f^{(l+1)}\|_F \leq C\rho\Theta\|\mathbf{Z}_f^{(l)}\|_F + C\Omega\|\mathbf{X}_k\|_F. \tag{20}$$

Since $\mathbf{Z}_f^{(0)} = \mathbf{0}$, and $\|\mathbf{W}_{i,f}^{(l)}\|_2 \leq \Omega$, on recursion we have,

$$\|\mathbf{Z}_f^{(l)}\|_F \leq C\Omega\sum_{j=0}^{l-1}(C\rho\Theta)^j\|\mathbf{X}_k\|_F \tag{21}$$

$$\leq \kappa\|\mathbf{X}_k\|_F, \tag{22}$$

where

$$\kappa = C\Omega\sum_{j=0}^{L-1}(C\rho\Theta)^j.$$

Further we simplify the recurrence, to do so consider $E^{(l+1)} = \left\|\hat{\mathbf{Z}}_f^{(l+1)} - \mathbf{Z}_f^{(l+1)}\right\|_F$, we have

$$E^{(l+1)} \leq C\kappa\epsilon\Theta\|\mathbf{X}\|_F + C\Theta\rho E^{(l)} \tag{23}$$

Now expanding above recursion with $l = 0, 1, 2, ..$ we have

$$\begin{aligned} E^{(1)} &\leq C\kappa\epsilon\Theta\|\mathbf{X}\|_F \\ E^{(2)} &\leq C\kappa\epsilon\Theta\|\mathbf{X}\|_F + C^2L\epsilon\Theta^2\rho\kappa\|\mathbf{X}\|_F \\ E^{(L)} &\leq C\kappa\Theta\left(\sum_{p=0}^{L-1}(C\rho\Theta)^p\right)\epsilon\|\mathbf{X}\|_F, \\ &\vdots \\ \left\|\hat{\mathbf{Z}}_f^{(L)} - \mathbf{Z}_f^{(L)}\right\|_F &\leq \text{const}\epsilon\|\mathbf{X}\|_F, \end{aligned} \tag{24}$$

where $\text{const} = C\kappa\Theta\left(\sum_{l=0}^{L-1}(C\rho\Theta)^l\right)$ and equation 24(a), follows since the initialization being same. It can be observed that the error in the representations only scale linearly with energy of the perturbation matrix ϵ. Recall this is the error for ARMA-1 recursion after L iterations. However, for ARMA-F the error remains the same, since

$$\begin{aligned} \left\|\hat{\tilde{\mathbf{Z}}} - \tilde{\mathbf{Z}}\right\|_F &\leq \frac{1}{F}\sum_{f=1}^{F}\left\|\hat{\mathbf{Z}}_f^L - \mathbf{Z}_f^L\right\|_F \\ &\leq \text{const}\epsilon\|\mathbf{X}\|_F \end{aligned} \tag{25}$$

Then error in the visit node embeddings are bounded as

$$\begin{aligned} \left\|\hat{\tilde{\mathbf{z}}}^{(v)} - \tilde{\mathbf{z}}^{(v)}\right\|_F &= \left\|\mathbf{e}_v^T\hat{\tilde{\mathbf{Z}}}_f - \mathbf{e}_v^T\tilde{\mathbf{Z}}_f\right\|_2 \\ &\leq \text{const}\epsilon\|\mathbf{X}\|_F, \end{aligned} \tag{26}$$

where recall $\mathbf{e}_v$ is a canonical vector. Leveraging this the error between the SSM representations for a current visit be bounded as

$$\begin{aligned} \|\hat{\mathbf{s}}_{i,k} - \mathbf{s}_{i,k}\|_2 &= \left\|\hat{\tilde{z}}_{i,k}^{(v)}\mathbf{B}^T\mathbf{G} - \tilde{z}_{i,k}^{(v)}\mathbf{B}^T\mathbf{G}\right\|_2 \\ &\overset{(a)}{\leq} \text{const}\gamma\delta\epsilon\|\mathbf{X}\|_F, \end{aligned} \tag{27}$$

where equation 27(a) follows since $\|\mathbf{B}\|_2 \leq \gamma$ and $\mathbf{G} = \|(\exp(\Delta_k\mathbf{A}) - \mathbf{I})\mathbf{A}^{-1}\| \leq \delta$. Therefore for a fixed visit it can be observed that the error in the proposed SSM representations scales only linearly with the energy of perturbation matrix ϵ. In what follows we present the proof accounting for the temporal error propagation. □

B.4 Proof for Corollary 1

Proof. We extend the analysis by explicitly unrolling the SSM dynamics. Recall that the difference between perturbed and unperturbed states satisfies:

$$\begin{aligned} \|\hat{\mathbf{s}}_{i,1} - \mathbf{s}_{i,1}\|_2 &\overset{(a)}{=} \|(\hat{\mathbf{s}}_{i,0} - \mathbf{s}_{i,0})\exp(\Delta_1\mathbf{A}) + (\hat{\tilde{\mathbf{z}}}_{\mathrm{i,1}}^{(\mathrm{v})} - \tilde{\mathbf{z}}_{\mathrm{i,1}}^{(\mathrm{v})})\mathbf{B}^{\mathrm{T}}\mathbf{G}\|_2 \\ &\overset{(b)}{\leq} \text{const}\gamma\delta\epsilon\|\mathbf{X}_{i,1}\|_F \end{aligned} \tag{28}$$

where (b) is realized by leveraging the result of Theorem 1 and by observing that initializations remain the same. For $k = 2$, we have

$$\begin{aligned} \|\hat{\mathbf{s}}_{i,2} - \mathbf{s}_{i,2}\|_2 &\overset{(a)}{=} \|(\hat{\mathbf{s}}_{i,1} - \mathbf{s}_{i,1})\exp(\Delta_2\mathbf{A})\|_2 + (\hat{\tilde{\mathbf{z}}}_{\mathrm{i,2}}^{(\mathrm{v})} - \tilde{\mathbf{z}}_{\mathrm{i,2}}^{(\mathrm{v})})\mathbf{B}^{\mathrm{T}}\mathbf{G}\|_2 \\ &\overset{(b)}{\leq} \eta\text{const}\gamma\delta\epsilon\|\mathbf{X}_{i,1}\|_F + \text{const}\gamma\delta\epsilon\|\mathbf{X}_{i,2}\|_F \end{aligned} \tag{29}$$

where (b) is realized through $\|\exp(\Delta_k \mathbf{A})\|_2 \leq \eta$. Also observe that the error consists of (i) propagated error from the previous step and (ii) newly injected perturbation:

Generalizing this for any $k = T$, we have

$$\begin{aligned} \|\hat{\mathbf{s}}_{i,T} - \mathbf{s}_{i,T}\|_2 &\leq \text{const}\gamma\delta\epsilon \sum_{\tau=1}^{K} \eta^{K-\tau} \|\mathbf{X}_{i,\tau}\|_2 \\ &\overset{(b)}{\leq} \text{const}_1.\epsilon \end{aligned} \tag{30}$$

Here $\text{const}_1 = \text{const}\gamma\delta \sum_{\tau=1}^{\mathrm{K}} \eta^{\mathrm{K}-\tau} \|\mathbf{X}_{\mathrm{i},\tau}\|_{\mathrm{F}}$. This expression explicitly captures the temporal residual propagation, where perturbations introduced at earlier time steps are attenuated (or propagated) through the factor $\eta^{K-\tau}$.

Importantly, the above result shows that the error remains linear in the perturbation magnitude ϵ, and the proportionality constant now explicitly captures both layerwise and temporal cascading effects. □

B.5 Proof of Theorem 4.2

Proof. Recall the normalized hypergraph adjacency operator is defined as

$$\mathbf{A}_{H,f} = \mathbf{D}_{v,f}^{-1/2} \mathbf{\Xi}_{H,f} \mathbf{W} \mathbf{D}_{e,f}^{-1} \mathbf{\Xi}_{H,f}^{\top},$$

where $\mathbf{\Xi}_{H,f}$ denotes the incidence matrix, and $\mathbf{D}_{v,f}$ and $\mathbf{D}_{e,f}$ are the node and hyperedge degree matrices, respectively.

Under a permutation of node labels induced by $\mathbf{P}$, the adjacency matrix modifies as

$$\mathbf{A}_{H,f}^{\text{perm}} = \mathbf{P}\mathbf{A}_{H,f}\mathbf{P}^{\top}.$$

Similarly, the node feature matrix is permuted as $\mathbf{X}_f^{\text{perm}} = \mathbf{P}\mathbf{X}_f$.

For notational convenience, we drop the reliance of i, k. The hypergraph ARMA computes node representations as

$$\mathbf{Z}_f^{(l+1)} = \sigma\left(\mathbf{A}_{H,f}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l+1)} + \mathbf{X}_k\mathbf{W}_{i,f}^{(l+1)}\right)$$

Under the permuted node labels and inputs, representations modifies as

$$\begin{aligned} \mathbf{Z}_{f,\text{perm}}^{(l+1)} &\overset{(a)}{=} \sigma\left(\mathbf{A}_{H,f}^{\text{perm}}\mathbf{Z}_{f,\text{perm}}^{(l)}\mathbf{W}_{r,f}^{(l+1)} + \mathbf{X}_{k,perm}\mathbf{W}_{i,f}^{(l+1)}\right) \\ &\overset{(b)}{=} \sigma\left(\mathbf{P}\mathbf{A}_{H,f}\mathbf{P}^{\top}\mathbf{P}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l+1)} + \mathbf{P}\mathbf{X}_k\mathbf{W}_{i,f}^{(l+1)}\right) \\ &\overset{(c)}{=} \mathbf{P}\sigma\left(\mathbf{A}_{H,f}\mathbf{Z}_f^{(l)}\mathbf{W}_{r,f}^{(l+1)} + \mathbf{X}_k\mathbf{W}_{i,f}^{(l+1)}\right) \\ &\overset{(d)}{=} \mathbf{P}\mathbf{Z}_f^{(l+1)}, \end{aligned} \tag{31}$$

where the equation 31(c) follows from the orthogonality of permutation matrices and the fact that $\sigma(\cdot)$ is applied row-wise. equation 31(d) establishes that representations from ARMA hypergraph neural networks are permutation equivariant. For ARMA-F based approach, the representations remain the permutation equivariant since

$$\begin{aligned} \tilde{\mathbf{Z}}_{\text{perm}} &= \frac{1}{F}\sum_{f=1}^{F} \mathbf{Z}_{f,\text{perm}}^{L} \\ &= \frac{1}{F}\sum_{f=1}^{F} \mathbf{P}\tilde{\mathbf{Z}}_f^{L} \\ &= \mathbf{P}\tilde{\mathbf{Z}} \end{aligned} \tag{32}$$

Recall, the input to the dynamic state-space model is the embedding of the visit node, extracted as

$$\tilde{\mathbf{z}}_{\text{perm}}^{v} = \mathbf{e}_{v,\text{perm}}^{\top}\tilde{\mathbf{Z}}_{\text{perm}},$$

where $\mathbf{e}_v \in \mathbb{R}^{|\mathcal{V}_{i,t}|}$ is the canonical basis vector corresponding to the visit node. Under permutation, the node representations are reordered as in equation 31, yielding

$$\begin{aligned}\tilde{\mathbf{z}}^v_{\text{perm}} &= \mathbf{e}_v^\top \mathbf{P}^T \mathbf{P}\tilde{\mathbf{Z}}. \\ &\overset{(a)}{=} \mathbf{e}_v^\top \tilde{\mathbf{Z}}, \\ &= \tilde{\mathbf{z}}^v \end{aligned} \tag{33}$$

where (a) follows from the property of permutation matrix. Since the proposed model operates solely on the visit-level representations although ARMA introduces the permutation equivariance at the node level, visit node selection induces invariance hence the output from SoAR-SSM are invariant to permutations. □

C ARMA Realization of the Sobolev Hypergraph Operator

To establish the connection between the Sobolev filtering operator and its recurrent realization, consider the Sobolev filter operator,

$$\mathbf{R}_{H,k} = \left[\mathbf{I}_N + \lambda(\bar{\mathbf{L}}_{H,k} + \epsilon\mathbf{I}_N)^\beta\right]^{-1} \tag{34}$$

$$\mathbf{R}_{H,k} \overset{(a)}{=} \left[\mathbf{I}_N + \lambda\left(\mathbf{I}_N - \mathbf{L}_{H,k} + \epsilon\mathbf{I}_N\right)\right]^{-1} \tag{35}$$

$$= \left[(1 + \lambda(1+\epsilon))\,\mathbf{I}_N - \lambda\mathbf{L}_{H,k}\right]^{-1}, \tag{36}$$

where equation 36(a) follows with $\beta = 1$ and by normalizing $\bar{\mathbf{L}}_{H,k}$ such that $\mu_{\max} = 2$ and $\mu_{\min} = 0$. Further, let us define

$$\alpha_0 = \frac{\lambda}{1 + \lambda(1+\epsilon)}, \qquad \alpha_1 = \frac{1}{1 + \lambda(1+\epsilon)} \tag{37}$$

we obtain

$$\mathbf{R}_{H,k} = \alpha_1\left(\mathbf{I}_N - \alpha_0\mathbf{L}_{H,k}\right)^{-1} \tag{38}$$

Realize that the response of the Sobolev filter matches the ARMA-1 filter response (Bianchi et al., 2021). Leveraging this connection, now we can implement give a recursive update to implement such Sobolev filter. In particular, similar to ARMA-1 filter, Sobolev filter is realized as

$$\mathbf{Z}^{(l+1)} = \alpha_0\mathbf{L}_{H,k}\mathbf{Z}^{(l)} + \alpha_1\mathbf{X}_k \tag{39}$$

Applying the recursion, we have ,

$$\mathbf{Z}^{(L)} = (\alpha_0\mathbf{L}_{H,k})^L\mathbf{Z}^{(0)} + \alpha_1\sum_{j=0}^{L-1}(\alpha_0\mathbf{L}_{H,k})^j X_k \tag{40}$$

As $L \to \infty$, the geometric series converges, yielding

$$\mathbf{Z}^\star = \alpha_1\left(I_N - \alpha_0\mathbf{L}_{H,k}\right)^{-1}\mathbf{X}_k = \mathbf{R}_{H,k}\mathbf{X}_k \tag{41}$$

This establishes the connection of Sobolev filter to ARMA-1 recursion.

D Implementation Details

D.1 Hyperknowledge Graph Construction

The construction of the hyperknowledge graph is performed in two sequential stages, as illustrated in Figure 5.

Global HyperKnowledge Graph Construction: To ensure semantic consistency across the EHR dataset, we create global hyperknowledge graph in two phases inspired by TnT-LLM architecture (Wan et al., 2024).

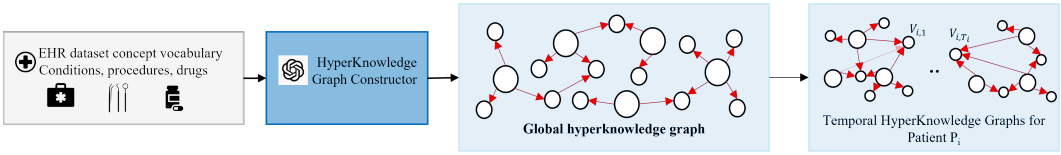

Figure 5: Our pipeline to construct hyperknowledge Graphs

Table 5: Examples of global hyperedges constructed during the taxonomy discovery phase.

Example Global Hyperedges	
Gastrointestinal Disease	**Heart Failure**
Example entities: • Agents for treatment of hemorhoids and anal fissures (topical) • Antacids • Antidiarrheal microorganisms • Gastritis and duodenitis • Nausea and vomiting • Gastrointestinal hemorrhage	*Example entities:* • Congestive heart failure; nonhypertensive • Diagnostic ultrasound of heart (echocardiogram) • Swan-ganz catheterization for monitoring • Beta blocking agents • Cardiac glycosides • Potassium-sparing diuretics

- Taxonomy (Phase 1): We first induce a global medical taxonomy by processing a representative subset of the corpus. This stage corresponds to the taxonomy generation phase in TnT-LLM, where a global clinical taxonomy is constructed. Using a diverse sample, we prompt the LLM to identify high-level clinical concepts that serve as global hyperedges or buckets (e.g., Heart Failure). These hyperedges constitute the foundation of the global hyperknowledge graph. The prompt used for clinical bucket generation is shown in Figure 6, and example hyperedges are presented in Table 5.
- Guided Batch (Phase 2): In this phase, we scale taxonomy assignment across full dataset while maintaining semantic consistency. Medical entities are processed in batches, where LLM is conditioned on cumulative set of existing hyperedges discovered in Phase 1. The LLM is instructed to reuse existing hyperedges whenever a semantic match exists and is allowed to introduce new hyperedges only when an entity represents fundamentally distinct clinical concept. Furthermore, recognizing the nature of clinical concepts, our prompting supports multi-label assignment (polysemy), this allows a single entity to map to multiple relevant hyperedges (e.g., an entity "Topical products for joint and muscular pain" is mapped to both "Bone and Joint Disease" and "Pain Management"). The prompt for this phase is shown in Figure 7 and examples are shown in Table 6.

D.1.1 Knowledge Infused Temporal Hypergraph Construction

Given a patient visit, we construct knowledge infused temporal hypergraphs by grounding the global hyperknowledge graph to the medical concepts observed in that visit. We first identify all medical concepts present in the visit and select only those hyperedges from the global hyperknowledge graph

Table 6: Examples of multi-label entity-to-hyperedge mappings from the guided batch assignment phase.

Medical Entity	**Mapped Global Hyperedges**
Topical products for joint and muscular pain	• Bone and Joint Disease • Pain Management
Selective calcium channel blockers with direct cardiac effects	• Hypertension • Cardiac Arrhythmias • Coronary Artery Disease

Role: You are a Senior Medical Ontologist and Knowledge Graph Architect working with ICU data. You have a dataset of *[Total Entities]* ICU medical entities (Conditions, Procedures, Drugs).

Your Task: Analyze the diverse random sample below and define a comprehensive set of **Medical Fact Buckets** that:

1. Cover these sample items
2. Would likely cover other similar ICU medical concepts in the full dataset
3. Are broad enough to group many entities but distinct enough to be meaningful

Core Principle:

- A bucket = a medical fact
- Everything clinically related to that fact belongs in it, whether it is a condition, a procedure, or a drug

Instructions:

1. The bucket name must be the medical fact itself
2. Create a reasonable number based on the clinical diversity you observe in the sample:
   - Not so few that unrelated concepts get merged together
   - Not so many that you recreate a flat list of individual entities
3. One entity can belong to **multiple buckets** if clinically relevant
   - Example: "metoprolol" in ["Hypertension", "Heart Failure", "Atrial Fibrillation"]
4. Use exact entity names
5. A bucket should focus on clinical relationships (what treats what, what manages what)

Sample Entities (*[Batch Size]* items):
- **Conditions:** *[Comma-separated list of conditions...]*
- **Procedures:** *[Comma-separated list of procedures...]*
- **Drugs:** *[Comma-separated list of drugs...]*

Output STRICT JSON format:

```
{
  "suggested_buckets": [
    "Heart Failure"
  ]
}
```

CRITICAL: Return ONLY the JSON object. Bucket names must be concise medical facts/concepts.

Figure 6: The prompt used for Taxonomy Discovery (Phase I), where the LLM induces clinically grounded medical fact buckets from a representative scout batch of ICU entities.

that contain at least one of these concepts. For each selected hyperedge, we retain only the concepts that actually appear in the visit.

In addition, we introduce a visit node and connect it to all selected hyperedges to explicitly model visit-level context. The resulting visit-level hypergraph is defined as

$$\mathcal{H}_{i,t} = (\mathcal{V}_{i,t}, \mathcal{E}_{i,t}), \tag{42}$$

where $\mathcal{V}_{i,t}$ denotes the set of unique clinical entities appearing in the visit $\mathcal{C}_{i,t}$, together with the visit node, and

$$\mathcal{E}_{i,t} = \{e_{i,t}^{(1)}, e_{i,t}^{(2)}, \ldots, e_{i,t}^{(M_{i,t})}\} \tag{43}$$

Role: You are a medical knowledge graph expert.
Task: Map the following *[Batch Size]* entities to one or more **Medical Concept Buckets**.
Core Principle:

- A bucket = a medical fact/concept
- Conditions, procedures, and drugs all belong in the same bucket if they are clinically associated with that fact

Consistency Constraint:

[Insert consistency instruction / existing buckets here]

Rules:

1. **REUSE:** If an entity fits into one of the EXISTING buckets above, you MUST use that exact name.
2. **CREATE:** Only create a NEW bucket name if the entity represents a distinctly new clinical concept not covered above.
3. **Polysemy:** An entity can belong to MULTIPLE buckets if it has distinct clinical uses.
   - Example: "beta blocking agents" $\rightarrow$ ["Management of Hypertension", "Management of Heart Failure"]
4. Bucket names must be concise medical concept nouns.
5. **Exhaustiveness:** You MUST output a key-value pair for EVERY SINGLE entity in the input list below.

Input Entities:

```
[
  "Aspirin",
  "Metoprolol",
  "Chest X-Ray",
  ... (list of entities in current batch)
]
```

Output STRICT JSON format:

```
{
  "entity_mappings": {
    "exact_entity_name_from_input": ["Bucket Name 1", "Bucket
        Name 2"],
    "another_entity_name": ["Bucket Name 3"]
  }
}
```

Important:

- The keys of the JSON object MUST be the exact strings from the input list - no typo fixes, no case changes.
- You MUST output a key-value pair for EVERY single entity in the input list.
- Count the input entities (*[Batch Size]*) and ensure the output JSON has the same number of keys.
- Values must be arrays/lists of bucket names, even if only one bucket.
- A bucket should focus on clinical relationships (what treats what, what manages what)

CRITICAL: Return ONLY the JSON object.

Figure 7: The updated prompt used for Phase II (Guided Batch Mapping), enforcing reuse of existing buckets and clinically grounded concept mapping.

represents the set of hyperedges associated with the visit. A single visit may be connected to one or multiple hyperedges, depending on the number of clinically related concept groups. Table 7 represents statistics for temporal hyperknowledge graphs.

Table 7: Statistics for MIMIC-III and MIMIC-IV Hypergraph Representations

Dataset	Task	Patients	Visits	Total Hyperedges	Hyperedges per Visit		
					Mean (SD)	Min	Max
MIMIC-III	Mortality Prediction	6,186	9,717	263736	27.14 (6.53)	4	33
	Length of Stay Prediction	35,707	44,399	1,383,419	31.16 (11.07)	3	69
	Drug Recommendation	35,707	44,399	1,204,642	27.13 (9.17)	3	57
MIMIC-IV	Mortality Prediction	59,262	132,275	3,277,198	24.78 (8.21)	3	54
	Length of Stay Prediction	123,478	232,247	6,711,737	28.90 (12.03)	2	93
	Drug Recommendation	46,184	154,953	3,281,934	21.18 (6.37)	2	44

Table 8: Training time comparison till convergence on MIMIC-IV mortality

Dataset	Model	Time (minutes)
MIMIC-IV	GraphCare	89
	KARE	285
	SoAR-SSM	**69**

D.2 Temporal Reasoning Paths

We adopt a gradient based attribution strategy that directly leverages the embeddings produced by ARMA graph neural networks, without introducing additional learnable parameters. Let $\mathbf{x}_{i,k}^{(j)} \in \mathbb{R}^D$ denote the embedding of clinical entity j for patient i at visit k. To quantify the contribution of this entity to the final prediction $\hat{y}_i$, we compute the gradient of $\hat{y}_i$ with respect to $\mathbf{x}_{i,k}^{(j)}$ i.e, $\nabla_{\mathbf{x}_{i,k}^{(j)}} \hat{y}_i$. This gradient reflects how changes in an entity's representation at a given visit propagate through the latent SSM dynamics and affect the final prediction. Leveraging the gradients we then compute the saliency score (Shrikumar et al., 2017) which accounts for feature presence and sensitivity. In particular the saliency score $Sc(e,k)$ for entity j of patient i at visit k is given by $Sc(e,k) = {\mathbf{x}_{i,k}^{(j)}}^T \cdot \nabla_{\mathbf{x}_{i,k}^{(j)}} \hat{y}_i$. A higher positive saliency score indicates that the corresponding clinical entity contributes more strongly toward increasing the outcome.

To translate saliency scores into an interpretable temporal reasoning path, we perform gradient-based backtracking using a set of *anchors*, defined as the top-ω entities in the final visit K with the highest saliency scores. For each anchor, we reconstruct a reasoning path by tracing backward through time and selecting the entity at visit k i.e., e_k^* which maximizes the saliency score $Sc(e,k)$.

D.3 Learnable Parameters

Complementing the parameter efficiency analysis in Figure 1, here we present the learnable parameters in `SoAR-SSM` framework. These only include input projection matrices for dimensionality alignment and the weight matrices inherent to `ARMA` layers. Then in the SSM module the only learnable parameters are the continuous input matrix $\mathbf{B}$ and the output matrix $\mathbf{C}$. Since the state transition matrix $\mathbf{A}$ is HiPPO initialized (not learnable). Finally decoder involves learnable projection layer.

D.4 Computational Complexity and Run times for SoAR-SSM

Computation Complexity of SoAR-SSM Recall, the complexity of obtaining the representations for `SoAR-SSM` is dominated by two operations (a): ARMA hypergraph module (b): Temporal SSM module. Let T denote the number of visits, N the average number of nodes per visit. S denote average number of node–hyperedge incidences per visit, hidden dimension d, F parallel ARMA stacks and L recursion depth. Assuming sparse incidence-based hypergraph propagation, one ARMA recursion complexity is $\mathcal{O}(Sd + Nd^2)$. Whereas, the complexity for L recursive updates with F parallel stacks and T visits is given as $\mathcal{O}(TFL(Sd + Nd^2))$. The temporal SSM operation requires $\mathcal{O}(d^2)$ operations per visit and hence $\mathcal{O}(Td^2)$ over the complete visit sequence. The overall forward complexity is therefore $\mathcal{O}(TFL(Sd + Nd^2) + Td^2)$.

Runtime comparison We present the runtime comparison with the baselines in Table 8. The parameter efficiency of SoAR-SSM also translates into reduced computational time. In particular, SoAR-SSM achieves substantially lower runtimes than GraphCare, a representation-learning-based approach, highlighting its improved computational efficiency.

D.5 Frequency Response plots of ARMA hypergraph filter

In this section, we investigate how accurately the proposed ARMA recursion approximates its corresponding analytical spectral response. Specifically, we consider the normalized analytical response parameterized by $\alpha_0 = 0.7, 0.15$ and $\alpha_1 = 0.99, 0.1$. We then set $\mathbf{W}^{(l)}_{r,f} = \alpha_0$ and $\mathbf{W}^{(l)}_{i,f} = \alpha_1 \mathbf{1}_D$, and evaluate the recurrence in equation 6 without applying the non-linearity. Following the methodology of Bianchi et al. (2021), we compute the resulting system response for different recursion depths, i.e., $l \in 1, 2, 3$.

As shown in Fig. 8, increasing the recursion depth l progressively improves the approximation to the analytical (Ideal) response. In this example, the analytical response corresponds to a low-pass spectral filter. This experiment therefore demonstrates that, for a fixed set of parameters, successive ARMA recursions can effectively approximate the desired low-pass spectral response.

More generally, different ARMA branches can learn distinct low-pass responses by adopting different filter parameters, as illustrated in Fig. 9 labeled as branch. Leveraging the ARMA-F formulation, multiple such responses can subsequently be combined to construct a richer set of spectral filters, including low-pass, high-pass and band-pass responses, as shown in Fig. 9. Consequently, the proposed filtering mechanism is not restricted to a single spectral regime; rather, it enables the model to capture complementary spectral components across multiple scales.

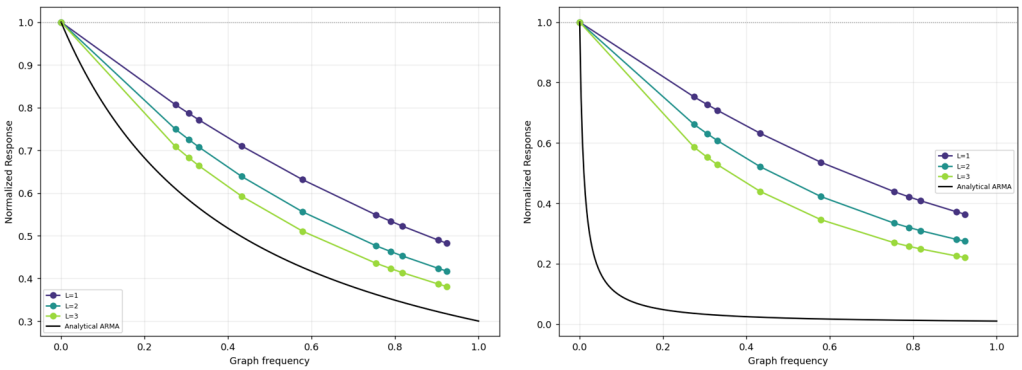

Figure 8: Spectral responses with varying depth

Dirichlet energy curves analysis on oversmoothing: To empirically show that the proposed model mitigates the oversmoothing we compute dirichlet energy with the increase in number of layers (l) and $F = 1$. Further, we compare it with its counter part hypergraph convolution (Hconv) built on using message passing operation. In Figure 10, we show the variation of normalized Dirichlet energy with recursion depth together with the distribution of spectral energy across low, mid, and high frequency components at each recursion depth. From the Dirichlet energy plot, we observe that the energy does not monotonically decay toward zero as the recursion depth increases as can be seen in Hconv. Instead, the learned representations retain a substantial fraction of their initial Dirichlet energy even at larger recursion depths whereas for Hconv it gradually decreases toward zero. This indicates that repeated ARMA propagation does not progressively collapse the node representations into an increasingly smooth hypergraph signal. The spectral energy distributions provide a complementary view of this behavior. With the increase in the recursion depth the mid and high frequency components remain non-negligible across the different recursion depths.

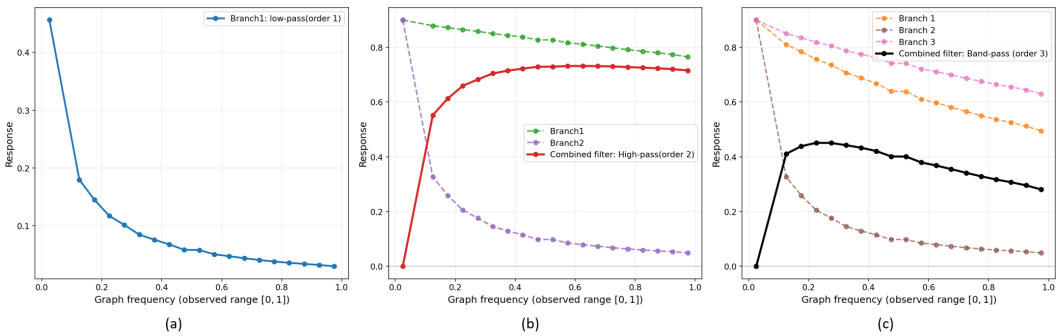

Figure 9: Frequency response of Hypergraph F-ARMA filter.

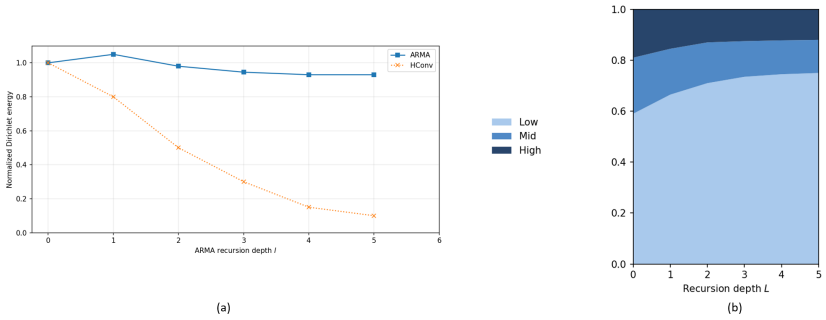

Figure 10: Energy versus the layer depth

D.6 Computation Overhead of Generating Hyperedges

Recall, the proposed method avoids naive per-patient hypergraph construction by employing a two-phase design optimized for scalability and reduced cost. Specifically, a global hyperknowledge graph is constructed once as an offline preprocessing step. In particular, global hyperknowledge graph construction for MIMIC-III, incurs a runtime as 178 sec and token cost as 100k (total tokens) $\times$ per token cost, while the per-patient hypergraph construction incurs runtime as 0.2 milliseconds. Importantly, this global construction is not repeated during training or inference, making it efficient in practice. Furthermore, when deploying in a new setting with expanded vocabulary, the system only processes the newly observed terms against the existing taxonomy, ensuring that additional token and computational costs remain controlled.

E Dataset and Training Details

E.1 Dataset Description

We use publicly available MIMIC-III (Johnson et al., 2016) and MIMIC-IV (Johnson et al., 2021) datasets. Table 9 summarizes the key statistics of datasets used in our experiments.

E.2 Task Description

We evaluate our model on four tasks on the electronic health record (EHR) data. Each task is formulated based on a patient's visit sequence.

- **Mortality Prediction.** This binary classification task predicts patient survival status during the next visit. The model uses a patient's previous visit records to output a binary prediction indicating survival status. During dataset preparation the final visit of each patient is excluded from training since it does not have a future outcome.

- **Length of Stay Prediction.** This task estimates the duration of a patient's intensive care unit (ICU) stay for a given hospital visit. It is formulated as a multi-class classification problem with 10 discrete categories. The target label is represented as a one-hot vector indicating the corresponding class among 10 categories, which represent ICU stays of less than one day (class 0), within one week (classes 1-7), between one and two weeks (class 8), and longer than two weeks (class 9).
- **Drug Recommendation.** This task predicts the set of medications prescribed to a patient during a hospital visit. The model outputs a set of drugs selected from a predefined medication vocabulary. Since multiple drugs may be prescribed simultaneously, this task is treated as a multi-label classification problem.
- **Phenotyping task.** The task involves identifying a patient's clinical conditions or disease phenotypes from EHR of the patient. In particular, the goal is to predict one or more phenotype labels from a predefined phenotype set. To do so, we formulate phenotyping as a multi-label classification problem, where the model outputs a probability for each phenotype and is evaluated based on its ability to accurately identify the clinical conditions associated with the patient.
- **Diagnosis Prediction.** For a patient's historical disease trajectory represented by a sequence of visits and corresponding diagnosis codes diagnosis prediction task involves predicting the set of disease codes that will appear in the patient's next visit. The task requires modeling both temporal evolution and inter-disease relationships to capture the underlying progression patterns of chronic and acute conditions.

Table 9: Statistics of EHR datasets. "#": "the number of", "/patient": "per patient".

Dataset	#patients	#visits	#visits/patient	#conditions/patient	#procedures/patient	#drugs/patient
MIMIC-III	35,707	44,399	1.24	12.89	4.54	33.71
MIMIC-IV	123,488	232,263	1.88	21.74	4.70	43.89

Baselines: We compare the proposed model against representation learning approaches. In particular we compare against GRU (Chung et al., 2014), Transformer (Vaswani et al., 2017), RETAIN (Choi et al., 2016b), GRAM (Choi et al., 2017), Deepr (Nguyen et al., 2017), StageNet (Gao et al., 2020), AdaCare (Ma et al., 2020), GRASP (Zhang et al., 2021) and GraphCare (Jiang et al., 2024), KARE (Jiang et al., 2025),HypEHR (Xu et al., 2023),HypKG (Xie et al., 2025),MEDCo (Kerdabadi et al., 2026a),LINKO (Nayebi Kerdabadi et al., 2025b).

Evaluation Metrics: Model performance is evaluated using a comprehensive set of metrics designed to capture distinct aspects of predictive reliability. **AUROC** (Area Under the Receiver Operating Characteristic Curve), **AUPRC** (Area Under the Precision-Recall Curve), **F1 Score**, **Accuracy**, **Jaccard**, and **Cohen's Kappa**.

We use binary cross-entropy loss for mortality. Cross-entropy loss for drug recommendation and length of stay prediction.

E.3 Data Preprocessing

Both MIMIC-III and MIMIC-IV datasets are split into train, validation and test splits with 80%/10%/10% ratio. For dataset preprocessing we followed the similar setup as in (Jiang et al., 2024; 2025).

To ensure that the evaluation is strictly inductive, all dataset splitting and representation-construction steps are performed without using information from validation or test patients during training. Patients are partitioned into disjoint training, validation, and test sets, and all visit prefixes belonging to a patient remain within the same split. Therefore, no patient history is shared across splits. All data-dependent preprocessing is fitted using the training split only. In particular, the global hyperknowledge taxonomy is induced from the training vocabulary and then kept fixed when constructing validation and test hypergraphs. Concepts that are not observed during training are assigned using predefined type-level fallback groups. Likewise, any co-occurrence statistics or clustering used for alternative hypergraph construction are estimated exclusively from training visits and subsequently applied unchanged to validation and test data. This prevents held-out visit statistics from influencing the learned representation space.

For drug recommendation, medications are treated exclusively as prediction targets. Medication entities are excluded from the hypergraph construction for all visits, and hyperedges that contain no remaining condition or procedure entities after masking are removed. Visit-level structural statistics are also computed only after this masking step. Consequently, the input hypergraph is determined solely by the observed conditions and procedures and does not encode either the identities or structural counts of the medications to be predicted.

Finally, each visit-level hypergraph is constructed independently from the information available for that visit, and no message passing is performed across patients. Node features are frozen biomedical language-model embeddings of concept names and do not depend on dataset-level co-occurrence statistics. Thus, under this protocol, validation and test information affects neither model training nor the construction of training representations.

E.4 Training Details

We use Adam as the optimizer, patience as 85, runs as 10 on MIMIC-III and 5 on MIMIC IV. The detailed hyperparameters used for training are summarized in Table 10,

Table 10: Hyperparameter configuration for SoAR−SSM across tasks on MIMIC-III and MIMIC-IV. Mort: Mortality. Drug: Drug Recommendation

Hyperparameter	**MIMIC-III**			**MIMIC-IV**		
	Drug	**LOS**	**Mort.**	**Drug**	**LOS**	**Mort.**
Learning rate (lr)	0.001	0.001	0.001	0.001	0.001	0.001
Hidden dimension	384	256	128	384	128	128
SSM state dimension	384	128	128	64	128	128
Batch size	128	128	128	64	128	128
Dropout	0.1	0.0	0.2	0.1	0.2	0.3

E.5 System Configuration

All experiments were conducted on a server equipped with two 64-core Intel Xeon Platinum 8562Y+ CPUs with 512GB memory. The platform runs on Ubuntu 22.04.5 LTS with GCC version 10.5.0. We used CUDA 11.8, Pytorch version 2.1.2 and Pytorch-geometric 2.7.0 for all the experiments. All experiments were performed on a single NVIDIA A40 GPU with 44GB of VRAM.

F Additional Experiments

F.1 Ablation based on source of hyperknowledge creation

To evaluate the robustness of our framework to the source of domain knowledge, we include additional experiments where hyperedges are constructed without any LLM involvement. Specifically, we consider (i) ontology-based grouping using standard medical ontologies where the individual medical entities are mapped to broad, standardized category labels (e.g., "Circulatory System Diseases") using the CCS and ATC ontologies to build global hyperedges. Then, for each patient visit, specific hypergraph is constructed by activating only the predefined hyperedges corresponding to the entities present in that patient's visits, formally linking the concepts and a central visit node to these hyperedges via an incidence matrix and (ii) co-occurrence-based grouping, where concept co-occurrence frequencies are first computed across all patient visits and normalized using Positive Pointwise Mutual Information (PPMI). The resulting concept relationships are then clustered into 50 groups for drug recommendation task and 100 groups for all other tasks, each forming a hyperedge, creating global hyperknowledge graph. For each patient visit, specific hypergraph is subsequently instantiated by activating only the hyperedges corresponding to concepts present in that record. As shown in Table 11, SoAR−SSM with these non-LLM constructions achieves performance that is comparable to SoAR−SSM where hyperedges are built using LLMs. Further, we emphasize that performance of this approach is still significantly better than, prior state-of-the-art methods. While LLM-based hyperedges provide additional semantic refinement, the *core gains persist even with*

static or data-driven constructions, indicating that the improvements primarily stem from modeling higher-order relations via hypergraphs and long-range temporal dependencies via SSM. Overall, these results demonstrate that SoAR-SSM is robust to the choice of hyperedge construction and remains effective even in resource-constrained settings without LLMs.

Table 11: SoAR-SSM performance comparison with different hyperedge construction strategies on MIMIC-III across mortality prediction, LOS prediction, and drug recommendation tasks.

Task	Metric	Without LLM		With LLM
		Unsupervised Clustering	Medical Ontologies	**SoAR-SSM**
Mortality	AUPRC	31.33 ± 0.50	35.39 ± 1.26	$\mathbf{36.10 \pm 1.30}$
	AUROC	71.71 ± 0.80	75.84 ± 0.19	$\mathbf{76.20 \pm 1.00}$
LOS	Kappa	27.54 ± 0.30	30.12 ± 0.10	$\mathbf{31.30 \pm 0.70}$
	F1	35.20 ± 0.40	38.52 ± 0.20	$\mathbf{39.44 \pm 0.60}$
Drug	Jaccard	40.71 ± 1.30	51.90 ± 0.70	$\mathbf{53.27 \pm 0.50}$
	F1	55.17 ± 1.50	66.10 ± 0.20	$\mathbf{67.90 \pm 0.90}$

F.1.1 Calibration metrics on mortality prediction

To evaluate the robustness of performance gains, we report calibration metrics for mortality prediction task on MIMIC-III in Table 12. It can be observed that SoAR-SSM achieves lower ECE and Brier score, along with higher specificity and sensitivity, indicating more reliable and well-calibrated predictions. We emphasize compared to prior state-of-the-art methods, SoAR-SSM demonstrates *improved robustness, further supporting our claims*.

Table 12: Calibration metrics

Method	ECE	Brier Score	Sensitivity	Specificity
GraphCare	–	–	0.17	0.97
KARE	–	–	0.14	0.94
SoAR-SSM	0.04	0.10	0.18	0.98

F.2 Comparison with KARE

As shown in Table 13, SoAR-SSM consistently outperforms KARE in both zero-shot and few-shot settings for mortality prediction task on MIMIC-III. While the fine-tuned variant of KARE achieves the highest Macro-F1, it depends on task-specific LLM fine-tuning and careful hyperparameter optimization. More importantly, fine tuned variant incurs significant computational and token costs as shown in runtime and parameter efficiency comparisons. In contrast, SoAR-SSM attains strong performance without any LLM fine-tuning, highlighting its effectiveness on EHR prediction tasks.

Table 13: Macro F1 (%) comparison of SoAR-SSM and KARE on MIMIC-III and MIMIC-IV for mortality prediction.

Model	MIMIC-III
KARE (Zero-shot)	54.6
KARE (Few-shot)	53.5
KARE (Fine-tuned)	64.6
SoAR-SSM	59.1

F.3 Results on Phenotyping and Diagnosis Prediction Task

In Table 14, we report the results on the phenotyping task to enable a fair comparison with prior hypergraph methods, namely HypEHR (Xu et al., 2023) and HypKG(Xie et al., 2025), which are commonly evaluated on this benchmark. The results show that the proposed model consistently outperforms existing approaches, highlighting the importance of jointly modeling higher-order relationships and temporal evolution. More importantly, it asserts the effectiveness of proposed method in modeling temporal hypergraphs, multi-frequency components and long-range dependencies for improved performance. Further, with diagnosis prediction as a downstream task, we also compare the performance of our model with the recent MKG based approaches LINKO(Nayebi Kerdabadi et al., 2025a) and MEDCO(Kerdabadi et al., 2026b). We report AUPRC and Acc@k metrics. Accuracy@k (Acc@k) evaluates whether at least one ground-truth diagnosis code is included among the model's top-k predicted diagnoses. It can be observed from Table 15, compared to KG based approaches temporal hypergraph based modeling together with multi-frequency modeling and long range temporal propagation enables in improved efficiency.

Table 14: Performance comparison on phenotyping task.

Method	AUROC	F1 Score
HypEHR	82.19 ± 0.13	41.51 ± 0.52
HypKG	84.26 ± 0.17	45.30 ± 0.49
SoAR-SSM	88.54 ± 0.04	46.54 ± 0.04

Table 15: Diagnosis Prediction performance comparison on MIMIC-III and MIMIC-IV.

Dataset	Model	AUPRC	Acc@15	Acc@20	Acc@30
MIMIC-III	LINKO	44.91	52.30	55.62	62.80
	MEDCo	47.21	54.20	57.40	64.39
	SoAR-SSM	**48.30**	**55.49**	**59.77**	**70.05**
MIMIC-IV	LINKO	48.14	57.78	61.21	67.97
	MEDCo	50.00	59.22	62.54	68.71
	SoAR-SSM	**53.62**	**61.10**	**65.73**	**74.18**

F.4 Performance Across Different LLM Backbones

To examine whether the performance of SoAR-SSM depends on the choice of LLM used within the framework, we evaluate SoAR-SSM with three different LLM backbones: GPT-4.1, Gemini 2.5 Pro, and Claude Sonnet 4. As shown in Table 16, SoAR-SSM maintains consistent performance across all three LLMs. These results indicate that the effectiveness of SoAR-SSM is not tightly coupled to a particular LLM backbone. Instead, the framework maintains consistent predictive performance across LLMs from different models, suggesting that its gains primarily arise from the proposed framework rather than from the choice of a specific LLM.

Table 16: Performance of SoAR-SSM with different LLMs on MIMIC-III mortality prediction task.

LLM Backbone	AUROC	AUPRC
GPT-4.1	76.1 ± 0.9	36.1 ± 1.3
Gemini 2.5 Pro	76.6 ± 0.8	37.7 ± 1.1
Claude Sonnet 4	77.4 ± 0.5	37.7 ± 1.1

G Reasoning Path Examples

In this section, we present additional examples (Fig 11 and Fig 13) of reasoning paths obtained using our gradient-based temporal reasoning procedure over `SoAR-SSM`. For each patient, we extract the top-2 most influential entities per visit via gradient attribution on the trained model, then prompt an LLM with the resulting visit-ordered entity trajectory to produce a natural-language explanation of the model's prediction. We include both a positive case (in-hospital mortality) and a negative case (survival). The prompt used for generating the explanation is given in figure 12.

H Limitations

The proposed framework depends on large language models (LLMs) for constructing hyperknowledge graphs, which can be susceptible to hallucinations and inconsistencies. Although the model demonstrates strong performance even with static hypergraph construction methods, leveraging LLMs provides additional accuracy gains and also provides hypergraph creation method for which ontologies are not clearly defined. However, this benefit comes at the cost of increased token usage and associated computational overhead.

I Future Directions

While this work focuses on healthcare applications, the proposed framework is general and can be extended to domains that require long-context reasoning over structured data, such as question answering and retrieval-augmented generation, where exploiting higher-order relational structure with SSMs is a promising direction.

Mortality label=0, Predicted label=0
Visit 1:
↪ **Entities:** *chronic ulcer of skin, aspiration pneumonitis; food/vomitus*
↓
Visit 2:
↪ **Entities:** *aspiration pneumonitis; food/vomitus, high-ceiling diuretics*
↓
Visit 3:
↪ **Entities:** *septicemia, aspiration pneumonitis; food/vomitus*
↓
Visit 4:
↪ **Entities:** *chronic ulcer of skin, septicemia*
LLM explanation: This patient presents with a chronic skin ulcer and recurring aspiration pneumonitis, which are evident from the first visit. The patient's condition is further complicated by the administration of high-ceiling diuretics in the second visit, potentially exacerbating fluid and electrolyte imbalances. In the third visit, the development of septicemia indicates a systemic infection, likely a complication of the recurring aspiration pneumonitis. Despite these complications, the chronic skin ulcer persists throughout the visits, suggesting a pre-existing condition contributing to the patient's vulnerability. The most recent visit shows a stabilization of the septicemia, which, combined with the persistent chronic skin ulcer, suggests a narrowly averted in-hospital mortality event.

Figure 11: Gradient based temporal reasoning path (negative case).

Based on the following reasoning paths generated for patient mortality prediction, provide a detailed textual explanation of what happened:

Reasoning Paths Data:
`{formatted_output}`

Please provide a clear, structured explanation of the reasoning paths, including:

1. What the paths represent
2. Key entities and relationships identified
3. How they relate to mortality prediction
4. Any notable patterns or insights
5. Give concise text explanation

Figure 12: The prompt template used to generate textual explanations from the reasoning paths. The `{formatted_output}` placeholder is replaced by the specific paths extracted from the TKG.

Mortality label=1, Predicted label=1
Visit 1:
↪ **Entities:** *septicemia, opioid analgesics*
↓
Visit 2:
↪ **Entities:** *acute and unspecified renal failure, anxiolytics*
↓
Visit 3:
↪ **Entities:** *other and unspecified benign neoplasm, opioid analgesics*
↓
Visit 4:
↪ **Entities:** *posterior pituitary lobe hormones, anxiolytics*
↓
Visit 5:
↪ **Entities:** *posterior pituitary lobe hormones, anxiolytics*
↓
Visit 6:
↪ **Entities:** *septicemia, opioid analgesics*
↓
Visit 7:
↪ **Entities:** *anxiolytics, other and unspecified benign neoplasm*
↓
Visit 8:
↪ **Entities:** *acute and unspecified renal failure, anxiolytics*
↓
Visit 9:
↪ **Entities:** *acute and unspecified renal failure, anxiolytics*
↓
Visit 10:
↪ **Entities:** *septicemia, opioid analgesics*
↓
Visit 11:
↪ **Entities:** *anxiolytics, opioid analgesics*
LLM explanation: The patient's clinical picture initially presents with septicemia and opioid analgesics use, which is followed by acute and unspecified renal failure and the use of anxiolytics in multiple visits. Concurrently, there is a noted presence of other and unspecified benign neoplasm and continued opioid analgesics use across several visits. Towards the end, the patient continues to experience renal failure and anxiety, with the reoccurrence of septicemia. The patient's trajectory suggests a complex and persistent clinical condition, with multiple comorbidities, that may require close monitoring and management.

Figure 13: Gradient based temporal reasoning path (positive case).