From Keys to Knowledge: A Retrieval Framework Leveraging Transformer Internal Memory

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) has emerged as a powerful approach for improving the factual accuracy of large language models (LLMs), particularly by mitigating hallucinations, incorporating up-to-date information, and enhancing generalization across domains. However, current RAG methods often suffer from limitations due to their reliance on extended input prompts and a dependency on supervised retrievers for external knowledge access. In this work, we introduce Keys-to-Knowledge (K2K), a novel retrieval framework that shifts the paradigm from external document retrieval to internal, key-based knowledge retrieval within the LLM itself. K2K employs lightweight knowledge infusion to encode essential information directly into the model's parameter space, enabling the use of its internal key-value memory for retrieval. To improve the quality of query representations, we propose an activation-guided probe construction method. Furthermore, we introduce a cross-attention reranking mechanism to extract diverse and relevant information from the model's enriched internal knowledge. Experimental results on health outcome predictions demonstrate that K2K significantly improves both the efficiency and effectiveness of knowledge-intensive tasks, offering a promising alternative to traditional RAG approaches by eliminating the need for external retrieval pipelines. [1]

1 Introduction

Large Language Models (LLMs) have demonstrated strong performance across a wide range of natural language processing (NLP) tasks, such as link prediction, question answering, text classification, etc (Li and Ji, 2022; Achiam et al., 2023; Li et al., 2024a; Guo et al., 2025). However, a fundamental limitation remains: it is challenging for LLMs to incorporate newly emerging knowledge beyond their static pre-training data. To address this, retrieval-augmented generation (RAG) has emerged as an effective solution (Lewis et al., 2020; Li et al., 2025), enabling LLMs to dynamically retrieve relevant information from external corpora, thereby enhancing performance on downstream tasks.

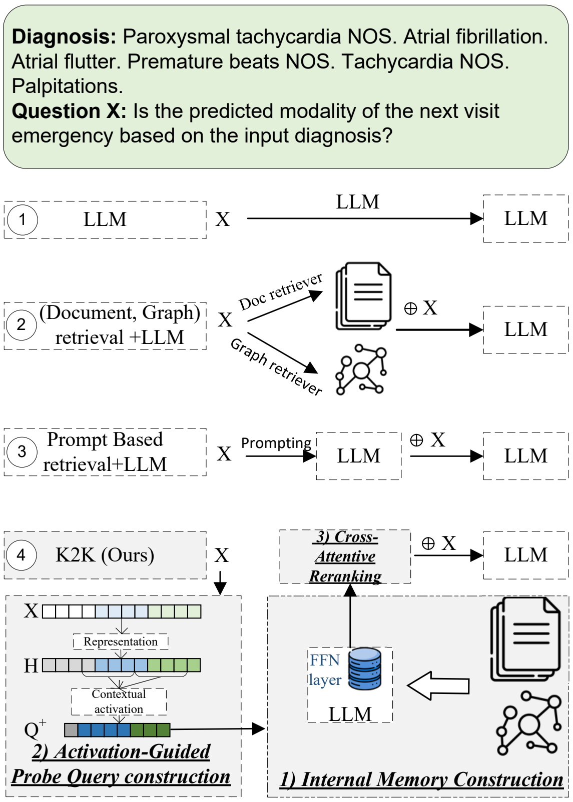
Diagnosis: Paroxysmal tachycardia NOS. Atrial fibrillation. Atrial flutter. Premature beats NOS. Tachycardia NOS. Palpitations.
Question X: Is the predicted modality of the next visit emergency based on the input diagnosis?
1
LLM
X
LLM
LLM
2
(Document, Graph) retrieval +LLM
X
Doc retriever
Graph retriever
⊕ X
LLM
3
Prompt Based retrieval+LLM
X
Prompting
LLM
⊕ X
LLM
4
K2K (Ours)
X
3) Cross-Attentive Reranking
⊕ X
LLM
X
Representation
H
Contextual activation
Q+
2) Activation-Guided Probe Query construction
FFN layer
LLM
1) Internal Memory Construction

Figure 1: Comparison of retrieval-augmented generation pipelines and our proposed K2K Approach.

Existing studies have explored various aspects of the RAG pipeline, with most efforts focusing on knowledge retrieval from structured sources (e.g., knowledge graphs, Wikidata) (Li and Huang, 2023; Zhang et al., 2025), unstructured documents (Jin et al., 2025). As shown in Figure 1, Pipelines 2 and

[1] The code is available here: `https://anonymous.4open.science/r/K2K-2390/README.md`

3. While these approaches have improved RAG performance, recent studies (Su et al., 2025) have highlighted two main limitations. **First**, Injecting knowledge through input prompts inevitably increases the context length, especially for tasks with already long inputs. **Second**, training a high-quality retriever remains a challenging task. Developing customized retrieval modules typically requires a large number of query-context pairs for supervised training, which imposes substantial demands on labeled data and computational resources, especially when dealing with massive, heterogeneous knowledge sources.

As prior work (Geva et al., 2020) has demonstrated, the Keys in the feed-forward layers (FFN) of transformer-based language models implicitly store factual knowledge. These Keys correspond to the vectors of the first projection matrix in the FFN, representing semantic units. One potential direction to address the above limitations is to retrieve these Keys as a source of internal knowledge. This approach bypasses the need to inject a long external context, thereby avoiding excessive input length, and also eliminates the dependence on external retrievers that require supervised training, since the knowledge is accessed directly from the model's own parameters.

However, using the query without incorporating contextual activation signals to retrieve top-k Keys do not guarantee accurate and relevant retrieval from the knowledge space. In our preliminary experiments, we observed that different queries often yield highly similar retrieved Keys, suggesting that the resulting probe query representations exhibit low discriminative power. In particular, these representations tend to obscure important semantic distinctions, ultimately leading to less effective knowledge retrieval. A similar observation was also reported by (Xiao et al., 2025). Otherwise, directly retrieving knowledge from the LLM's internal key space by top-k strategies lacks interpretability and structural awareness, as the retrieved key vectors are latent and not grounded in explicit sources like documents or knowledge graphs. Moreover, the retrieval process is static and non-adaptive, lacking explicit semantic signals to guide the reweighting of the retrieved knowledge.

To solve these issues, we propose **Keys-to-Knowledge (K2K)**, a novel retrieval framework that directly retrieves key-based knowledge from LLMs infused with external information. The framework consists of three main components: Internal Memory Construction, Activation-Guided Probe Query Construction, and Cross-Attentive Reranking, as shown in Figure 1, the numbered circle 4.

More specifically, 1) We construct a retrieval memory from the pre-trained language model. For knowledge not present in the pre-training corpus, we apply LoRA (Hu et al., 2021) to adapt the model and inject new knowledge. The Keys stored in the FFN layers collectively form this retrieval memory. This mitigates the reliance on external retrievers and alleviates the burden caused by long input contexts. 2) To effectively estimate the important tokens during inference and recognize the scarce outlier features, we construct the probe-query[2] for each context window to retrieve the relevant knowledge from retrieval memory, and designate activated query vectors with prominent activation bias to dominate the representation of probe-query for accurate retrieval, where the activation bias is computed by a diagonal approximation of the Mahalanobis distance between each token and the mean token to balance per-dimension variance. 3) Due to the varying relevance and structural dependencies across different knowledge, as well as the need for dynamic, context-dependent integration, we introduce a cross-attentive reranking mechanism that dynamically integrates multi-source knowledge conditioned on the query.

2 Preliminaries

2.1 Feed-Forward Layers as Unnormalized Key Memories

Feed-forward Layers In transformer-based architectures (Vaswani et al., 2017), the feed-forward network (FFN) operates alongside the self-attention mechanism and plays a crucial role in representations. Each feedforward layer is a position-wise function, processing each input vector independently. Given an input vector $\mathbf{x} \in R^d$, typically obtained from the attention layer, the output of the feed-forward layer FF(.) can be formulated as:

$$\mathrm{FF}(\mathbf{x}) = W_2 \cdot f(\mathbf{x} \cdot W_1) \tag{1}$$

To align this with the key-value memory (Geva et al., 2020), we can define :

$$\mathrm{FF}(\mathbf{x}) = V \cdot f(\mathbf{x} \cdot K^\top) \tag{2}$$

[2] A probe query is a representation derived from the current input that is used to retrieve relevant information from the model's internal key space.

where $K, V \in R^{d_m \times d}$ are learnable weight matrices,$K^\top = W_1$ and $V = W_2$, and $f(\cdot)$ denotes an activation function such as ReLU.

Feed-forward Layers in Lora To incorporate specific knowledge to the LLM, we follow the low-rank adaptation (LoRA) formulation by introducing trainable matrices $A \in \mathbb{R}^{h \times r}$ and $B \in \mathbb{R}^{r \times k}$, such that the FFN becomes:

$$\begin{aligned} \mathrm{FF}(\mathbf{x}) &= (W_2 + \Delta W_2) \cdot f(\mathbf{x} \cdot (W_1 + \Delta W_1)) \\ &= (W_2 + A_2 B_2) \cdot f(\mathbf{x} \cdot (W_1 + A_1 B_1)) \end{aligned} \tag{3}$$

where $W \in \mathbb{R}^{h \times k}$ is the original pre-trained weight matrix, and $f(\cdot)$ denotes an activation function such as ReLU. where, $K^\top = W_1 + A_1 B_1$ and $V = W_2 + A_2 B_2$.

3 Methodology

As illustrated in Figure 2, our K2K has three stages (1) Retrieval memory construction, (2) Activation-guided probe query construction for knowledge matching, and (3) a cross-attention reranking method is used to retrieve the key knowledge.

3.1 Retrieval Memory Construction

In our work, the retrieval memory primarily consists of two types of information: (1) document knowledge and (2) graph knowledge. To construct the memory from the document level, we begin by leveraging a pretrained large language model ($\mathcal{M}_{\text{base}}$) as the backbone. A common approach to encoding domain-specific document knowledge into an $\mathcal{M}_{\text{base}}$ is through continued pretraining. As an alternative to costly continued pretraining, we adopt an existing domain-adapted model ($\mathcal{M}_{\text{domain}}^{\text{doc}}$).

To adapt the graph information, we first convert each triple in the graph into its corresponding textual description, such as *the relationship between head entity and tail entity is relationship*. We then apply LoRA-based continued training to train $\mathcal{M}_{\text{domain}}^{\text{doc}}$ on the organized triples dataset, enabling it to encode domain-specific knowledge from the graph. After that, $\mathcal{M}_{\text{domain}}^{\text{doc}}$ is further adapted with graph information and becomes $\mathcal{M}_{\text{domain}}^{\text{doc+graph}}$.

We use the Keys from the FFN within l-th Transformer layer of $\mathcal{M}_{\text{domain}}^{\text{doc}}$ as the internal representation of document-level knowledge, denoted as K_{doc}^{l}, same as the W_1 in equation (1). Similarly, we treat the LoRA adapter matrices $A_1 B_1$ (as shown in equation (3)) from the FFN layer of $\mathcal{M}_{\text{domain}}^{\text{doc+graph}}$ as the structured knowledge source derived from the knowledge graph in layer l, denoted as K_{graph}^{l}.

3.2 Activation-Guided Probe Query Construction

As suggested on (Xiao et al., 2025), existing probe queries often rely on widely used mean pooling strategies, which fail to capture the core semantics of the question. Their attention is dispersed across all tokens, rather than focusing on meaningful anchors. This limits their effectiveness for KV retrieval and motivates the need for a more semantically grounded query construction. To solve this issue, in our work we propose an Contextual Activation Weight to distinguish the importance of each query vector within a window context.

For query vector $H_t = [h_1^t, h_2^t, \ldots, h_w^t]$, where t refers the t-windows, w refers the token length of window t. We first calculate the statistical mean $\bar{z}^t$ in the window w,

$$\bar{z}^t = \frac{1}{w} \sum_{j=1}^{w} h_j^t \tag{4}$$

The previous work (Xiao et al., 2025) uses Euclidean distance to compute the weight of each token, but it suffers from the limitation of treating all dimensions equally, ignoring per-dimension variance and thus being less sensitive to meaningful deviations in low-variance directions (Weinberger and Saul, 2005; Xing et al., 2002). To address the limitations of Euclidean distance, we propose using a diagonal approximation of the Mahalanobis distance to better account for per-dimension variance. Unlike the full Mahalanobis distance, our approach avoids expensive matrix inversion, significantly reducing computational complexity and runtime.

$$\phi_j^t \approx \sqrt{\sum_{d=1}^{D} \frac{(h_{j,d}^t - \bar{z}_d^t)^2}{\sigma_d^2}} \tag{5}$$

In this formula, ϕ_j^t measures how much the j-th token deviates from the mean across each dimension, normalized by the variance σ_d^2, where d indexes the feature dimensions, σ_d^2 denotes the variance of the token representations along the d-th dimension within the context window.

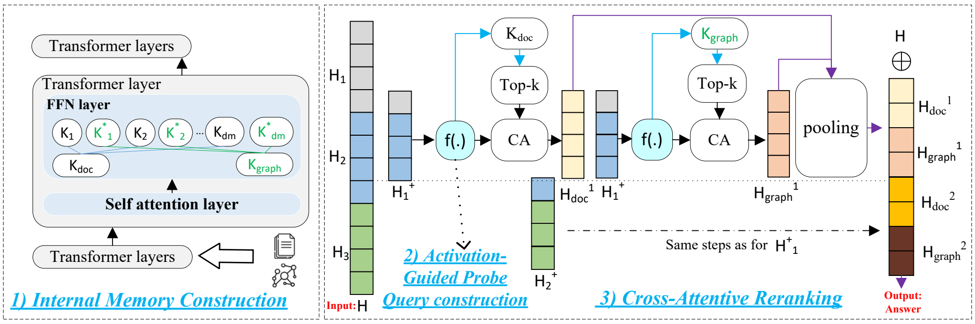

Figure 2: Overview of the K2K framework, consisting of three steps: (1) Retrieval Memory Construction builds $K_{\text{doc}} = [K_1, K_2, \cdots, K_{dm}]$ and $K_{\text{graph}} = [K_1^*, K_2^*, \cdots, K_{dm}^*]$; (2) Activation-Guided Probe Query Construction (Blue box function f(.)) enhances the query representation for key retrieval from K_{doc} and K_{graph}; (3) Cross-Attentive Reranking retrieves relevant document knowledge H_{doc} and graph knowledge H_{graph} for the enhanced query $Q_t^+ = f(H_t^+)$, and integrates them with the original representation H for final prediction. Here, $t \in 1, 2, 3$.

In the next, we normalize the activation bias scores ϕ_j^t across all tokens within the context window to obtain token-level weights α_j^t, ensuring that their sum equals 1. This allows us to treat the scores as a soft attention distribution:

$$\alpha_j^t = \frac{\phi_j^t}{\sum_{k=1}^{w} \phi_k^t} \tag{6}$$

In the last, we compute the enhanced probe vector Q_t for the context window by aggregating all token vectors h_j^t using the normalized weights α_j^t. This results in a single representation that emphasizes semantically important tokens:

$$Q_t = f(H_t) = \sum_{j=1}^{w} \alpha_j^t \cdot h_j^t \tag{7}$$

Here, $f(.)$ refers to the activation-guided probe query construction function described in this part.

3.3 Cross Attention Reranking

To perform cross-attention reranking, following RETRO (Borgeaud et al., 2022), we first split the representation H of input sentence into a sequence of $t-1$ chunks, denoted as $\{H_1^+, H_2^+, \ldots, H_{t-1}^+\}$. H_t^+ represents the query embeddings constructed by concatenating the last token of chunk C_t and the first $w-1$ tokens of chunk C_{t+1}. For each chunk C_t, we compute its contextualized query representation $Q_t^+ = f(H_t^+)$ using the probe query construction method introduced in Section 3.2. Next, we retrieve relevant knowledge for each chunk from two distinct knowledge sources: **Document Knowledge:** K_{doc}^l and **Graph Knowledge:** K_{graph}^l in the layer l. To construct the relevant document knowledge $K_{\text{doc}}^{\text{t}}$ and graph knowledge $K_{\text{graph}}^{\text{t}}$, we compute similarity scores between the query representation Q_t^+ and K_{doc}^l, K_{graph}^l, respectively. The top-k most relevant vectors are selected via:

$$\begin{aligned} K_{\text{doc}}^t &= \text{top-}k\left(\text{sim}(Q_t^+, K_{\text{doc}}^l)\right) \\ K_{\text{graph}}^t &= \text{top-}k\left(\text{sim}(Q_t^+, K_{\text{graph}}^l)\right) \end{aligned} \tag{8}$$

We apply Cross-Attention (CA) to rerank and select the most relevant document knowledge H_{doc}^t and graph knowledge H_{graph}^t for the query representation,

$$\begin{aligned} H_{\text{doc}}^t &= \text{CA}(Q_t^+, K_{\text{doc}}^{\text{t}}, V_{\text{doc}}^{\text{t}}) \\ H_{\text{graph}}^t &= \text{CA}(Q_t^+, K_{\text{graph}}^{\text{t}}, V_{\text{graph}}^{\text{t}}) \end{aligned} \tag{9}$$

Each knowledge is first processed by a pooling function $P(.)$ to normalize the vector dimensionality, after which they are fused through concatenation.

$$H_k^t = [P(H_{\text{doc}}^t); P(H_{\text{graph}}^t)] \tag{10}$$

We then aggregate all chunk-level fused representations together with the input sentence representation H and feed the combined representation into an MLP for final prediction. The loss is defined as:

$$\mathcal{L}_{\text{cls}} = \text{CrossEntropy}\left(\text{MLP}([H;\ H_k^1; H_k^2; \ldots; H_k^{t-1}]),\ y\right) \tag{11}$$

where y denotes the ground truth label.

4 Experiments

4.1 Testbeds Setup

In this work, we use healthcare prediction as our testbed, where relevant information is sparsely distributed and implicitly expressed within long and complex clinical events. This setting poses significant challenges for retrievers, as it requires capturing dispersed and subtle clinical signals that are not explicitly stated. More specifically, given hospital visits $V = \{v_1, v_2, ..., v_{|V|}\}$ for each patient, along with the associated International Classification of Diseases (ICD) codes C_i for each visit, the model aims to predict the patient's clinical outcome y_i (a binary label). Each visit v_i includes a list of ICD codes C_i, where each ICD code $c_i \in C_i$ represents a *code* and is associated with a name s_i in the form of a *short text snippet*. In our experiments, we consider two prediction tasks as testbeds: (1) Mortality prediction, where y_i indicates whether the patient dies in the subsequent visit v_{i+1}, and (2) Readmission prediction, which predicts if the patient will be readmitted into hospital within α days, same as KARE (Jiang et al., 2024), we set α=15.

4.2 Dataset

	III-Mort	III-Read	IV-Mort	IV-Read
Train	7,777	7,777	100,125	10,0125
Test	953	953	12,667	12,667
Dev	978	978	12,547	12,547

Table 1: Datasets Statistics, Mort refers to the Mortality. III refers to the MIMIC-III. Read refers to readmission.

We use the publicly available MIMIC-III (Johnson et al., 2016) and MIMIC-IV (Johnson et al., 2020) datasets. Table 1 presents statistics of the processed dataset. Both datasets are split into training, validation, and test sets in a 0.8/0.1/0.1 ratio grouped by patient and controlled with a fixed random seed (42). We ensure that all samples from the same patient are assigned to a single subset, with no overlap among the training, validation, and test instances, thereby preventing data leakage. Unlike KARE (Jiang et al., 2024), which randomly selects a subset of samples from MIMIC-IV, we use the entire dataset as our testbed to more closely reflect real-world clinical settings.

4.3 Baselines and Evaluation Metrics

Our baselines include several machine learning-based models: GRU (Chung et al., 2014), RETAIN (Choi et al., 2016), Deepr (Nguyen et al., 2016), AdaCare (Ma et al., 2020), StageNet (Gao et al., 2020), and TCN (Bai et al., 2018). We also compare against KARE (Jiang et al., 2025), the current state-of-the-art retrieval-based model for healthcare prediction tasks. In addition, we include standard RAG (Li et al., 2024b), which retrieves relevant patient examples to enhance model performance by using the Contriver (Izacard et al., 2021). Furthermore, we incorporate Prompt-Based Retrieval (Frisoni et al., 2024), which leverages in-context learning to instruct the LLM to generate relevant medical knowledge for prediction. Following Jiang et al. (2025, 2023b), we used F1, Jaccard, AUPRC, and AUROC as the evaluation methods. For implementation details, please refer to Appendix A.1.

4.4 Main Results

Table 2 presents the main results and highlights several key observations: (1) K2K consistently outperforms all other methods across all datasets and tasks. (2) Baseline retrieval methods fail to capture the semantic nuances of the input. Although KARE enhances retrieval by combining relevant documents with the shortest paths from the graph, such paths may overlook critical relational information. In contrast, our method retrieves key knowledge directly from the language model's internal knowledge store, enabling more comprehensive and context-aware retrieval. (3) We find that LLMs perform worse than traditional machine learning models when the input contains discontinuous or complex diagnoses and suffers from class imbalance between positive and negative samples. This is also observed by Gao et al. (2025). By introducing document-level knowledge and graph-based knowledge into the language model, our method achieves improved performance. For example, K2K outperforms LLMs without retrieval mechanisms on the mortality prediction task using the MIMIC-IV dataset. (4) We found that prompt-based retrieval outperforms standard RAG by retrieving knowledge from external documents, enabling the language model to generate more useful information that improves the classification results, as evidenced by improvements in AUPRC and AUROC on the Mortality-MIMIC-III dataset. (5) Although K2K only achieves the best Jaccard score on Readmission-MIMIC-IV, it consistently outperforms all baselines across the remaining metrics.

5 Analysis

To further evaluate the effectiveness of our framework, we conduct a series of analyses based on different components of our model. First, we investigate the impact of different knowledge sources by introducing two ablations: K2K without document knowledge and K2K without graph knowledge (Section 5.1). We also assess the perfor-

		Mortality-MIMIC-III				Readmission-MIMIC-III			
Type	**Model**	**F1**	**Jaccard**	**AUPRC**	**AUROC**	**F1**	**Jaccard**	**AUPRC**	**AUROC**
ML	GRU (Chung et al., 2014)	13.87	7.45	8.03	53.50	68.28	51.84	52.94	50.00
	RETAIN (Choi et al., 2016)	13.73	7.37	9.57	54.86	45.88	23.48	54.11	51.29
	Deepr (Nguyen et al., 2016)	13.87	7.45	7.58	51.66	68.28	51.84	51.68	49.70
	AdaCare (Ma et al., 2020)	12.90	6.89	7.80	50.69	63.49	46.51	52.83	52.27
	StageNet (Gao et al., 2020)	9.97	5.25	7.10	47.14	51.56	34.74	50.38	48.27
	TCN (Bai et al., 2018)	11.28	5.97	6.76	45.81	65.46	48.66	49.84	47.65
RAG Baselines	KARE (Jiang et al., 2025)	16.42	8.94	12.46	58.35	64.07	47.13	59.53	54.95
	Standard RAG (Li et al., 2024b)	15.92	8.65	10.40	57.84	63.03	46.02	57.70	51.34
Retrieval Modules (Same LLM)	w/o retriever	16.00	8.69	11.61	59.40	69.17	52.87	59.07	54.61
	KARE (Jiang et al., 2025)	18.01	9.90	9.72	56.65	61.64	44.55	56.67	50.97
	Standard RAG (Li et al., 2024b)	11.94	6.34	9.34	54.19	**69.73**	**53.52**	57.09	52.99
	Prompt Based Retrieval (Frisoni et al., 2024)	15.05	8.13	10.78	58.72	66.51	49.82	54.19	49.71
	K2K (Our Approach)	**18.55**	**10.22**	**15.22**	**61.05**	69.31	53.03	**62.49**	**56.64**

		Mortality-MIMIC-IV				Readmission-MIMIC-IV			
Type	**Model**	**F1**	**Jaccard**	**AUPRC**	**AUROC**	**F1**	**Jaccard**	**AUPRC**	**AUROC**
ML	GRU (Chung et al., 2014)	3.20	1.62	1.66	53.71	59.28	42.13	57.38	56.58
	RETAIN (Choi et al., 2016)	2.78	1.41	1.43	47.18	66.77	50.12	51.44	49.61
	Deepr (Nguyen et al., 2016)	2.86	1.46	1.57	51.48	**68.13**	51.66	52.27	50.44
	AdaCare (Ma et al., 2020)	2.98	1.52	1.53	51.41	47.96	31.54	52.12	50.38
	StageNet (Gao et al., 2020)	2.96	1.50	1.60	51.11	48.11	31.67	50.74	48.67
	TCN (Bai et al., 2018)	2.92	1.48	1.63	54.17	53.32	36.35	51.33	49.62
RAG Baselines	KARE (Jiang et al., 2025)	0.96	0.40	1.50	51.45	63.63	46.66	**69.10**	**67.31**
	Standard RAG (Li et al., 2024b)	5.66	2.91	2.71	65.93	63.14	46.14	66.60	65.74
Retrieval Modules (Same LLM)	w/o retriever	1.08	0.50	1.30	44.61	61.30	44.20	67.86	65.83
	KARE (Jiang et al., 2025)	1.33	0.67	1.46	49.55	61.75	44.67	67.09	65.44
	Standard RAG (Li et al., 2024b)	2.45	1.61	2.74	55.92	60.95	43.84	68.51	66.64
	Prompt Based Retrieval (Frisoni et al., 2024)	3.16	1.60	1.49	48.26	61.02	43.91	68.89	67.02
	K2K (Our Approach)	**6.61**	**3.42**	**2.93**	**66.50**	63.75	**46.79**	68.67	66.47

Table 2: Comparative analysis of various retrieval and machine learning models for mortality and readmission prediction tasks on the MIMIC-III and MIMIC-IV datasets. Following KARE (Jiang et al., 2025), we use the Mixtral-based model BioMistral-7B as the LLM backbone. ML refers to machine learning based methods

mance of directly using the LLM with its internal knowledge to make predictions, in order to validate the effectiveness of our key knowledge retrieval framework, which leverages cross-window attention (Section 5.2). Next, we compare different query representation strategies to demonstrate the effectiveness of our proposed diagonal approximation of the Mahalanobis distance (Section 5.3). Finally, we analyze the effect of retrieving knowledge from different LLM layers (Section 5.4).

For additional experiments on K2K, including the effect of different chunk sizes, the impact of the hyperparameter top-k in Equation 8, and analyses of retrieval and inference efficiency across various retrieval methods and pipelines, please refer to Appendix C,D,E, and F.

	Model	F1	Jaccard	AUPRC	AUROC
Mortality-III	K2K	18.55	10.22	**15.22**	**61.05**
	K2K w/o graph	**20.48**	**11.40**	13.18	60.54
	K2K w/o document	16.66	9.09	10.52	55.72
Mortality-IV	K2K Ours	**6.61**	**3.42**	**2.93**	**66.50**
	K2K w/o graph	4.50	2.30	2.51	60.86
	K2K w/o document	3.57	1.82	2.71	66.41
Readmission-III	K2K	69.31	53.03	**62.49**	**56.64**
	K2K w/o graph	**70.95**	**54.98**	60.87	54.55
	K2K w/o document	69.74	53.54	61.93	56.36
Readmission-IV	K2K Ours	**63.75**	**46.79**	**68.67**	**66.47**
	K2K w/o graph	55.31	38.23	66.14	64.06
	K2K w/o document	56.95	39.81	55.43	64.68

Table 3: Results of different knowledge sources in K2K

5.1 Impact of Different Knowledge Source

Table 3 presents the results of K2K using different knowledge sources. Specifically, K2K w/o document refers to the variant of K2K that uses only the retrieved graph knowledge K^t_{graph}, as described in Section 3.3. To ensure a fair comparison, the only difference between K2K and its ablated versions (w/o document or w/o graph) is the type of knowledge source used. From Table 3, we observe that the performance of K2K drops when either document or graph knowledge is removed, especially on the MIMIC-III dataset. Moreover, although K2K w/o graph achieves a higher F1 score, its lower AUPRC and AUROC suggest that it may overfit to a specific threshold and lacks robustness in distinguishing positive cases across varying decision boundaries. In contrast, K2K achieves more balanced performance across all metrics, indicating better generalization and retrieval effectiveness.

5.2 Direct Use vs. Retrieved Use of Pre-trained Knowledge

	Model	F1	Jaccard	AUPRC	AUROC
Mortality-III	K2K	**18.55**	**10.22**	**15.22**	**61.05**
	LLM	4.49	2.29	8.67	55.62
	LLM+Doc	16.00	8.69	11.61	59.40
	LLM+Graph	4.50	2.29	8.67	55.62
	LLM+Doc+Graph	16.00	8.70	11.61	59.41
Readmission-III	K2K	69.31	53.03	**62.49**	**56.64**
	LLM	64.10	47.17	60.81	54.57
	LLM+Doc	69.17	52.87	59.07	54.61
	LLM+Graph	44.31	28.46	56.57	48.87
	LLM+Doc+Graph	**70.81**	**54.81**	61.51	54.70
Mortality-IV	K2K	**6.61**	**3.42**	**2.93**	**66.50**
	LLM	2.05	1.03	1.59	51.64
	LLM+Doc	1.08	0.50	1.30	44.61
	LLM+Graph	3.24	1.60	1.52	50.08
	LLM+Doc+Graph	1.08	0.55	1.30	44.61
Readmission-IV	K2K	**63.75**	**46.79**	**68.67**	**66.47**
	LLM	60.06	42.92	66.15	64.64
	LLM+Doc	61.30	44.20	67.86	65.83
	LLM+Graph	48.97	32.43	50.80	48.30
	LLM+Doc+Graph	54.86	37.80	51.57	49.93

Table 4: Comparison of Knowledge-Enhanced Models on Mortality and Readmission Prediction (MIMIC-III/IV). LLM refers to Mixtral-7B. LLM+Doc denotes BioMixtral-7B, which is obtained by further training Mixtral-7B on a medical corpus. LLM+Graph refers to Mixtral-7B adapted to graph-based knowledge using LoRA. LLM+Doc+Graph represents BioMixtral-7B further adapted to graph knowledge via LoRA.

Table 4 shows the results of the experiments of different knowledge-enhanced models.We found that leveraging windowed cross-attention and Mahalanobis-guided query construction to retrieve internal key knowledge from the LLM yields superior performance compared to directly employing a knowledge-augmented LLM for downstream tasks. We guess the reason is that although knowledge augmented LLMs such as BioMixtral 7B encode medical knowledge through pretraining, they may not explicitly surface critical risk factors for specific knowledge. For instance, in the MIMIC-III mortality task, the model might miss the implication of structured features like *mechanical ventilation* or *high SOFA score* if not directly prompted. In contrast, our method retrieves relevant internal knowledge from the encoded medical graph, such as the relations between symptoms, interventions, and mortality and fuses it into the model input. This structured retrieval improves the model's ability to reason over clinical signals and enhances prediction accuracy.

	Model	F1	Jaccard	AUPRC	AUROC
Mortality-III	K2K	**18.55**	**10.22**	**15.22**	**61.05**
	K2K w Euclidean	16.97	9.27	9.67	57.25
	K2K (Mean Only)	12.06	6.42	8.45	52.51
Readmission-III	K2K	**69.31**	**53.03**	**62.49**	**56.64**
	K2K w Euclidean	63.27	46.28	58.26	53.25
	K2K (Mean Only)	63.98	47.03	54.67	50.92
Mortality-IV	K2K	**6.61**	**3.42**	**2.93**	**66.50**
	K2K w Euclidean	4.79	2.45	2.19	61.81
	K2K (Mean Only)	0.82	0.44	2.51	61.73
Readmission-IV	K2K	**63.75**	**46.79**	**68.67**	**66.47**
	K2K w Euclidean	63.56	46.59	67.87	66.41
	K2K (Mean Only)	56.26	39.14	67.71	65.58

Table 5: Comparison of K2K with different query construction methods.

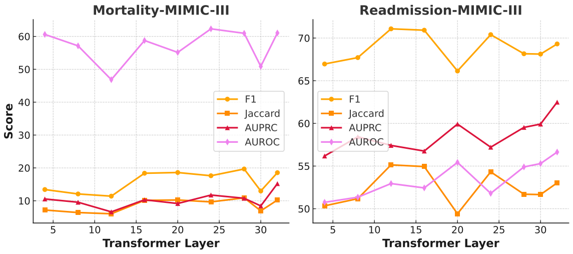

Figure 3: K2K performance with different layer knowledge. We used BioMistral-7B, which consists of 32 transformer layers.

5.3 Comparison of Query Representation Strategies

Table 5 presents various query representation strategies for assessing the importance of each query vector within a window context. K2K (Euclidean) uses Euclidean distance for token weighting, whereas K2K (Mean Only) computes the window representation via simple mean pooling. Table 5 shows that our Mahalanobis-guided query representation consistently outperforms prior approaches. Unlike Euclidean distance, which treats all dimensions equally, our method accounts for per-dimension variance and emphasizes informative low-variance directions. This leads to more precise token weighting and better contextual representations. The results validate the effectiveness of variance-aware distance metrics in enhancing retrieval-informed reasoning.

5.4 Comparison of Knowledge from Different LLM Layers

In this section, we conduct experiments on K2K using knowledge (key) sources stored in different transformer layers within the LLM. Both the document-based knowledge and the graph-based knowledge are extracted from the same corresponding layer. Figure 3 reveals a nuanced deviation from the conventional view that upper layers in Transformers primarily encode semantic features while lower layers capture shallow, surface-level patterns. Although the final layers (e.g., Layer 30+) do contribute positively to performance in both Mortality-MIMIC-III and Readmission-MIMIC-III tasks, this improvement is not strictly monotonic. Notably, several shallow layers (e.g., Layers 5, 8, and 10) also exhibit strong performance across multiple metrics, indicating that valuable structural or entity-level knowledge resides in the lower layers as well. Furthermore, the impact of each layer varies across different evaluation metrics (F1, Jaccard, AUROC), suggesting that knowledge is distributed in a non-linear fashion throughout the network. These findings underscore the importance of considering both shallow and deep layers in knowledge extraction and reasoning tasks.

6 Related Work

Many studies (Lewis et al., 2020; Guu et al., 2020; Li and Huang, 2023; Li et al., 2025; Jiang et al., 2025), have been proposed to use retrieved information from various knowledge stores to better understand the text or generate the expected output. For example, KIEST (Li and Huang, 2023) dynamically injects retrieved entity and attribute knowledge from the knowledge graph when generating the entity or attribute in the task of entity stage changes. REALM (Guu et al., 2020) employs a gradient-based method to reward the retriever, leading to improved prediction accuracy, while. KARE (Jiang et al., 2025) identifies relevant entities for each concept in the question and constructs a subgraph using the shortest paths between the retrieved entities and the query concept to provide structured relational context for downstream reasoning and answer generation. BiomedRAG (Li et al., 2025) employs a dynamic retrieval mechanism to rerank the initially retrieved top-k chunks from a constructed, diverse chunk database. RETRO (Borgeaud et al., 2022) proposes a chunk-based approach that uses attention mechanisms to rerank the retrieved top-k knowledge segments from an external knowledge base. To mitigate the challenges associated with injecting lengthy retrieved knowledge and to reduce retrieval latency from massive, heterogeneous knowledge sources, we propose a novel approach that retrieves knowledge directly from the key space of the LLM using a top-k and cross-window attention mechanism.

Recent work (Xiao et al., 2024; Liu et al., 2024; Fountas et al., 2025) has focused on designing retrieval modules that extract relevant information from the a key-value (KV) cache based on probe queries constructed from the current context tokens. These methods typically treat the current sliding window as a probe query to retrieve relevant key-value pairs from memory. However, most of these approaches overlook the importance of probe construction in the retrieval process, despite the fact that large language models (LLMs) are not inherently optimized for retrieval tasks. There are few works to explore how to construct the probe query in the key retrieval of the LLM. For example, the ActQKV (Xiao et al., 2025) proposes an activation-aware probe query mechanism that selects key tokens based on their activation methods and employs Euclidean distance to retrieve the most relevant key-value pairs. Nevertheless, this method assumes equal importance across all embedding dimensions, thereby ignoring per-dimension variance and reducing sensitivity to meaningful deviations in low-variance directions. This motivated us to develop a Mahalanobis-guided probe query construction method.

7 Conclusion

In this paper, we propose Keys-to-Knowledge (K2K), a novel retrieval framework that bypasses traditional external retrieval pipelines by leveraging the internal knowledge representations encoded within large language models. Unlike conventional RAG methods that rely on prompt-based input expansion, K2K retrieves relevant knowledge directly from the model's key space through a training-free, efficient mechanism. By incorporating Mahalanobis-guided query representation, and cross-window attention for dynamic multi-source integration, K2K demonstrates strong potential in enhancing reasoning and prediction in knowledge-intensive tasks. Our findings suggest that internal representations of LLMs are not only latent carriers of knowledge but can be explicitly accessed and utilized to improve performance without additional labeled data or costly retriever training.

8 Limitations

While our proposed K2K framework demonstrates strong performance in internal knowledge retrieval and integration, it still has several limitations. First, the retrieval memory is constructed from fixed layers of a pre-trained language model. Although the injected knowledge via LoRA enables domain adaptation, our current approach does not dynamically select which layers or representations (e.g., early vs. late FFN layers) are most informative for retrieval. Incorporating a layer-wise selection mechanism may further improve retrieval fidelity and efficiency. Second, our framework has been primarily evaluated within the biomedical domain. In the future, we plan to explore more challenging tasks and address the issue of data imbalance within these tasks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

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

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

Zafeirios Fountas, Martin A Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. 2025. Human-like episodic memory for infinite context llms. *arXiv preprint arXiv:2407.09450*.

Giacomo Frisoni, Alessio Cocchieri, Alex Presepi, Gianluca Moro, and Zaiqiao Meng. 2024. To generate or to retrieve? on the effectiveness of artificial contexts for medical open-domain question answering. *arXiv preprint arXiv:2403.01924*.

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

Yanjun Gao, Skatje Myers, Shan Chen, Dmitriy Dligach, Timothy Miller, Danielle S Bitterman, Guanhua Chen, Anoop Mayampurath, Matthew M Churpek, and Majid Afshar. 2025. Uncertainty estimation in diagnosis generation from large language models: next-word probability is not pre-test probability. *JAMIA open*, 8(1):ooae154.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Peng Jiang, Chang Xiao, Meng Jiang, and 1 others. 2025. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Pengcheng Jiang, Cao Xiao, Adam Cross, and Jimeng Sun. 2023b. Graphcare: Enhancing healthcare predictions with personalized knowledge graphs. *arXiv preprint arXiv:2305.12788*.

Pengcheng Jiang, Cao Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. 2024. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. *arXiv preprint arXiv:2410.04585*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2020. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, pages 49–55.

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

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems.

Mingchen Li and Lifu Huang. 2023. Understand the dynamic world: An end-to-end knowledge informed framework for open domain entity state tracking. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval.

Mingchen Li and Shihao Ji. 2022. Semantic structure based query graph prediction for question answering over knowledge graph. *arXiv preprint arXiv:2204.10194*.

Mingchen Li, Halil Kilicoglu, Hua Xu, and Rui Zhang. 2025. Biomedrag: A retrieval augmented large language model for biomedicine. *Journal of Biomedical Informatics*, 162:104769.

Mingchen Li, Chen Ling, Rui Zhang, and Liang Zhao. 2024a. Zero-shot link prediction in knowledge graphs with large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 753–760. IEEE.

Mingchen Li, Zaifu Zhan, Han Yang, Yongkang Xiao, Jiatan Huang, and Rui Zhang. 2024b. Benchmarking retrieval-augmented large language models in biomedical nlp: Application, robustness, and self-awareness. *arXiv preprint arXiv:2405.08151*.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, and 1 others. 2024. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *arXiv preprint arXiv:2409.10516*.

Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 825–832.

Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. 2016. Deepr: A convolutional net for medical records. arxiv. org.

Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. 2025. Parametric retrieval augmented generation. *arXiv preprint arXiv:2501.15915*.

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

Kilian Q Weinberger and Lawrence K Saul. 2005. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. 2024. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv e-prints*, pages arXiv–2402.

Qingfa Xiao, Jiachuan Wang, Haoyang Li, Cheng Deng, Jiaqi Tang, Shuangyin Li, Yongqi Zhang, Jun Wang, and Lei Chen. 2025. Activation-aware probe-query: Effective key-value retrieval for long-context llms inference. *arXiv preprint arXiv:2502.13542*.

Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. 2002. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

A Appendices

A.1 Implementation Detail

In this paper, we use Mistral-7B (Jiang et al., 2023a) as the $\mathcal{M}_{\text{base}}$ and employ BioMistral-7B (Labrak et al., 2024) as the $\mathcal{M}_{\text{domain}}^{\text{doc}}$. The chunk size is set to 64 throughout this work. For the top-k values, we use $k = 5$ for Mortality-MIMIC-III, $k = 20$ for Readmission-MIMIC-III and Mortality-MIMIC-IV, and $k = 10$ for Readmission-MIMIC-IV. The same LLM backbone is used during both the retrieval phase, when keys are extracted, and the training/inference phases, when those keys are utilized, ensuring alignment in the representation space. We use AdamW as our optimizer, with a learning rate of 2×10^{-5} and ϵ set to 1×10^{-8}. The batch size is 16. For the cross-attention module, we set the model dimension to 4096 and apply a dropout rate of 0.3.

A.2 Separately retrieval

We intentionally use only the base component W_1 from the final FFN layer of $\mathcal{M}^{\text{doc}}_{\text{domain}}$ to represent document knowledge. This design is motivated by the need to preserve a clear and interpretable separation between knowledge sources. Specifically, (1) **theoretically**, unstructured document knowledge (captured by W_1) and structured graph knowledge (injected via AB) differ fundamentally in format and reasoning mechanisms, and thus should not be merged directly in representation; (2) **in practice**, combining them into a single matrix $W_1 + AB$ would entangle their contributions, making it difficult to analyze or attribute model behavior to specific knowledge types; and (3) **from an engineering perspective**, separating the two enables more modular system design, facilitates ablation studies, debugging, incremental updates, and future knowledge extension.

B Mahalanobis distance

Step 1: Compute the Covariance Matrix Σ

$$\Sigma = \frac{1}{L-1} \sum_{j=1}^{L} (q_j^t - \bar{z}^t)(q_j^t - \bar{z}^t)^T \quad \in \mathbb{R}^{D \times D}$$

Step 2: Compute the Mahalanobis Distance (Activation Bias) ϕ_j^t

$$\phi_j^t = \sqrt{(q_j^t - \bar{z}^t)^T \Sigma^{-1} (q_j^t - \bar{z}^t)} \quad \in \mathbb{R}$$

Step 3: Construct the Probe-Query Vector $\mathbf{Q}^t_{\text{probe}}$

$$\mathbf{Q}^t_{\text{probe}} = \sum_{j=1}^{L} \alpha_j^t \cdot q_j^t, \quad \text{where} \quad \alpha_j^t = \frac{\phi_j^t}{\sum_{k=1}^{L} \phi_k^t} \quad \in \mathbb{R}^D$$

C Comparison of K2K with different chunk size

Figure 4 shows the K2K performance of different chunk sizes on the dataset MIMIC-III Mortality. We choose four chunk sizes: 16, 32, 64, and 128. We observe that smaller chunk sizes (e.g., 16) lead to higher F1 scores, indicating that finer granularity benefits the identification of relevant knowledge segments. However, chunk size 64 achieves the highest AUPRC and AUROC, suggesting it better balances precision and recall for more robust classification. Larger chunk sizes may reduce retrieval frequency but risk diluting critical signals. Therefore, chunk size selection should consider both task sensitivity and retrieval efficiency.

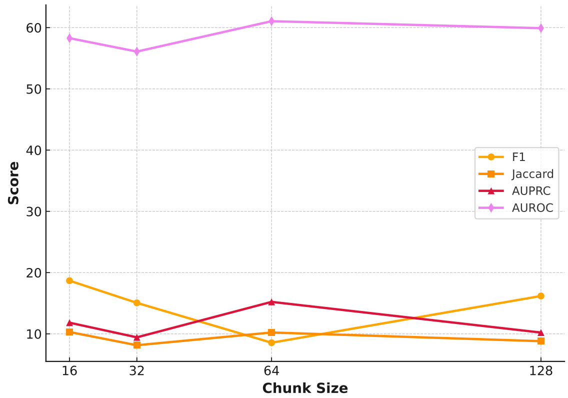

Figure 4: K2K performance with different chunk sizes.

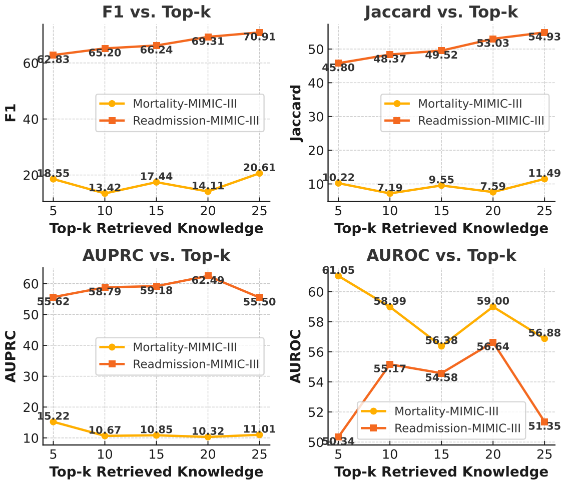

Figure 5: K2K Performance Across Different Top-k Retrieved Knowledge Values on MIMIC-III.

D Ablation Study on Top-k Retrieval

Figures 5 and 6 demonstrate how the number of retrieved knowledge entries (top-k) affects the performance of K2K on both MIMIC-III and MIMIC-IV datasets. For MIMIC-III, performance generally improves with increasing top-k, with the best F1 (20.61) and Jaccard (11.49) observed at k=25 for the mortality task, while the readmission task achieves optimal results at k=20–25. Notably, AUROC and AUPRC peak at k=20, suggesting a bal-

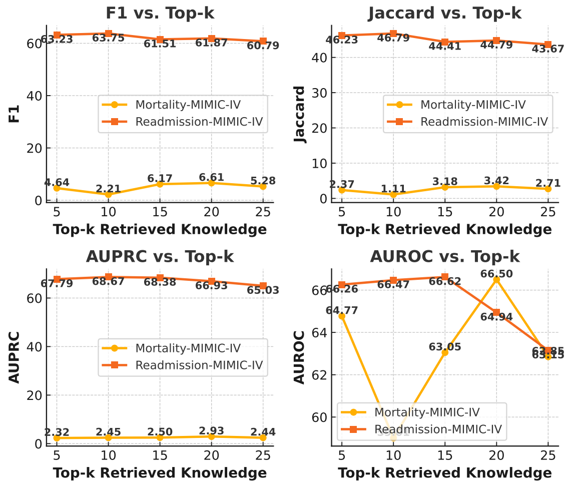

Figure 6: K2K Performance Across Different Top-k Retrieved Knowledge Values on MIMIC-IV.

ance between sufficient context and noise control.

In contrast, for MIMIC-IV, mortality prediction shows a performance peak at k=20 across all metrics, particularly for F1 and AUPRC, while readmission results are relatively stable across k values, with the highest F1 (63.75) and AUPRC (68.67) at k=10. However, large k values (e.g., k=25) tend to hurt AUROC, especially for readmission. These results indicate that task-specific tuning of top-k is crucial, and that mortality prediction benefits more from increasing top-k, while readmission may require a smaller, more focused knowledge set.

E Retrieval Efficiency Comparison Across different Retrieval methods

Compared to prior retrieval approaches, our K2K method demonstrates substantially higher efficiency. Specifically, KARE performs multi-stage reasoning by first retrieving co-existing concepts appear in each patient's data and then computing the shortest paths between the concepts and the co-existing concepts over a large knowledge graph. This results in a total complexity of $O(k(|V| + |E|))$, where k is the number of co-existing concepts, and $|V|, |E|$ are the number of nodes and edges in the graph, respectively. Contriever, a dense retriever, encodes the query and computes similarities across the entire corpus, resulting in a time complexity of $O(Nd)$ without approximation, where N is the number of documents and d is the embedding dimension. Prompt-based retrieval avoids external indexing but relies on LLM generation conditioned on carefully designed instructions, which incurs substantial inference cost at $O(L \cdot n^2 \cdot h)$, where L is the number of layers, n is the token length, and h is the number of attention heads.

In contrast, K2K bypasses both external document retrieval and graph traversal by directly reusing the internal knowledge of the LLM. It retrieves relevant knowledge by comparing current input representations with pre-trained FFN keys and LoRA adapter keys from a specific transformer layer. This enables fast memory access with a time complexity of only $O(m)$ or $O(mk)$ (for top-k selection), where m is the number of tokens. By removing the need for external retrieval or generation, K2K achieves the fastest inference speed among all retrievers while maintaining high accuracy, demonstrating the efficiency and practicality of internal knowledge utilization.

F Inference Efficiency Comparison Across Different Retrieval Pipelines

Compared with traditional RAG pipelines that rely on document or graph retrieval, our K2K framework significantly reduces inference cost by avoiding long-text prompting. In standard document retrieval, the retrieved contents are textual sequences that must be concatenated with the query as additional context, leading to an inference cost of $\mathcal{O}(N \cdot T \cdot d)$, where N is the number of retrieved documents, T is the average number of tokens per document, and d is the hidden dimension of the LLM. In contrast, K2K retrieves fixed-size internal memory vectors from the model itself, resulting in a much smaller cost of $\mathcal{O}(k \cdot d)$, where k is the number of keys retrieved, independent of token length. Furthermore, graph-based retrieval methods introduce an additional computational burden. Identifying relevant paths or subgraphs in a large-scale knowledge graph is typically NP-hard, which further increases the end-to-end latency. By retrieving and reweighting internal key vectors through cross-attention reranking, K2K bypasses both the inefficiencies of long-context prompting and the combinatorial complexity of graph path enumeration.