Technical Appendix: DualMAR: Medical-Augmented Representation from Dual-Expertise Perspectives

Anonymous submission

A. KG Construction

A.1 Data Sources in Healthcare KG.

To develop a comprehensive KG for studying diseases, we consider two primary resources related to diseases and phenotypes, and incorporate three additional resources to enhance the connectivity of our Diagnosis KG. Notably, these data resources—whether widely-used standardized ontologies or direct readouts from experimental measurements—primarily focus on extensive coverage across disease, phenotype, and drug entities. We begin with a brief introduction of primary data resources used to construct component of the Diagnosis KG:

1. *DrugBank.* DrugBank (Knox et al. 2024) is a comprehensive resource containing detailed pharmaceutical knowledge. We retrieved the latest version (5.1.12), published on March 14, 2024. Our focus is on synergistic drug interactions, which represent the bidirectional connections between two drugs.
2. *DrugCentral.* DrugCentral (Ursu et al. 2016) is a curated database that provides information on drug-disease interactions, including indications, contraindications, and off-label uses. We utilized the updated SQL Database, released on November 1, 2023, for our study.
3. *HPO.* The Human Phenotype Ontology (Gargano et al. 2024) offers detailed information on phenotype abnormalities associated with diseases. We used one of the most recent updates, from April 19, 2024, focusing on disease-phenotype and phenotype-phenotype relationships, and extracted verified associations with expertly curated annotations.
4. *ICD-9-CM Disease Ontology.* ICD-9-CM (Organization et al. 1988) is a coding system used by health insurers to classify medical conditions for billing purposes, and it underlies the diagnostic information in most EHR data. For our purposes, we extracted the parent-child relationships among codes to describe disease interactions.
5. *SIDER.* The Side Effect Resource (Kuhn et al. 2016) contains data on side-effect phenotype caused by various drugs. We retrieved this information from the data release dated October 21, 2015.

A.2 Data Sources in LLM-generated KG.

We also extract textual features for disease nodes in the KG from several widely-used clinical knowledge bases. The in-context learning capabilities of LLMs are employed to convert this textual information into triples, which are then integrated into the KG. The following four public data sources are used for triple generation:

1. *Mayo Clinic.* Mayo Clinic (Brennan, Miner, and Rizza 1998) is a nonprofit academic medical center and biomedical research institution that maintains a knowledge base with clinical information on over 2000 diseases. We collected Mayo Clinic web knowledge in 2023 and mainly focus on both "Symptoms & Causes" (Overview, Symptoms, When to see a doctor, Causes, Risk Factors, Complications, and Prevention) and "Diagnosis & Treatment" parts for each condition.
2. *Orphanet.* Orphanet (Weinreich et al. 2008) is a database dedicated to rare diseases. We collected data in 2023, focusing on information about definitions, prevalence, treatment, and clinical descriptions for rare diseases.
3. *Rare Disease Database.* Rare Disease Database (Schieppati et al. 2008), created by a nonprofit organization NORD, serves as a supplementary source for rare diseases. In 2023, we collected data to retrieve information on disease overviews, symptoms, causes, treatments, and clinical trials.
4. *Wikipedia.* Wikipedia is a free, open-content online encyclopedia maintained by volunteers worldwide. It includes an official list of ICD-9 condition codes. We web-scraped the overall hierarchy of ICD-9 codes and gathered related information for diseases with individual Wikipedia pages through valid hyperlinks. For example, the page for "Typhoid Fever" provides detailed information on symptoms, causes, diagnosis, prevention, treatment, epidemiology, history, terminology, and societal impact, offering more comprehensive and diverse knowledge compared to other textual sources.

A.3 Prompting LLM

It is important to note that since all textual information for disease nodes is sourced from either publicly verified or expertly curated data sources, issues of data quality or hallucination in the medical knowledge generated by the LLM are

Table 1: General Prompting Framework for Triple Generation Tasks of LLMs

Name	Prompt Template
Variables	<category>: The prompt is not only suitable for disease but also available for concepts within medical domains such as medication and treatment. In this study, only condition concepts are considered. <term>: The concept name. In this study, it means disease names provided by web-scraped text. <topics>: The topics related to crawled text, and it has been provided when we crawled them from websites. <text>: The content of crawled text.
Skeleton	Given a crawled text about specific topic of certain <category>, please find triples related to the given <category> in terms of crawled text. • Filling triples in updates based on given information and strictly following output style of example updates. • Each update should follow the format of [ENTITY 1, RELATIONSHIP, ENTITY 2] with directed edge. • Both ENTITY 1 and ENTITY 2 should be noun, and one of them must be <term>. • Just output each unique triple once, don't output repeatedly. • It is possible that <category> name not exactly matched in crawled text (abbreviated or partly matched), consider it as the same thing. Example: ## An example demo is shown below... Given a paragraph about specific topic of certain <category>, please find triples related to the given <category> in the text. **Given Information:** <category> **Name**: <term> **Topics**: <topics> **Text**: <text> **Updates:** ## LLM's output is expected to generate here in terms of given information... ## [Head Entity, Relation, Tail Entity]...
Example	**Disease Name:** Heart Failure **Topics:** Overview **Text:** *Heart failure occurs when the heart muscle doesn't pump blood as well as it should. When this happens, blood often backs up and fluid can build up in the lungs, causing shortness of breath. Certain heart conditions gradually leave the heart too weak or stiff to fill and pump blood properly. These conditions include narrowed arteries in the heart and high blood pressure. Proper treatment may improve the symptoms of heart failure and may help some people live longer. Lifestyle changes can improve quality of life. Try to lose weight, exercise, use less salt and manage stress. But heart failure can be life-threatening. People with heart failure may have severe symptoms. Some may need a heart transplant or a device to help the heart pump blood. Heart failure is sometimes called congestive heart failure.* **Updates:** • [Heart Failure, IS_CAUSED_BY, Narrowed Arteries], • [Heart Failure, IS_CAUSED_BY, High Blood Pressure], • [Heart Failure, HAS_SYMPTOMS, Shortness of Breath], • [Heart Failure, HAS_SYMPTOMS, Fluid Build-up in Lungs], • [Heart Failure, NEEDS_TREATMENT, Proper Treatment], • [Heart Failure, NEEDS_TREATMENT, Lifestyle Changes] ## You can use more examples to refine output of LLMs, but only one example is also fine for this task.

not a primary concern. In this section, we focus on how to utilize the in-context learning capabilities of LLMs to transform this textual information into structured triples that can be directly integrated into the KG.

The LLM used in this study is `Llama-3-8b`. Table 1 illustrates a carefully designed prompt for constructing triples from disease conditions. The primary objective is to harness the extensive knowledge embedded in the LLM to extract meaningful triples consisting of two entities and a relationship. `Llama-3-8b` was chosen for inference in this study for several reasons: first, its relatively small parameter size makes it easier to deploy and use; second, it is one of the latest and most widely adopted open-source models; and third, the task at hand is common and straightforward, not requiring the large parameter size or fine-tuning that other medical-domain LLMs might offer.

Our strategy begins with a prompt related to a specific medical condition. The LLM is then tasked with generating a list of triples, following the format `[ENTITY 1, RELATIONSHIP, ENTITY 2]`, where both `ENTITY 1` and `ENTITY 2` are nouns. The goal is to generate these triples with both breadth (capturing potential abbreviations or partial matches) and accuracy (ensuring that each triple is based on existing information).

Although triple generation is relatively straightforward for LLMs, it still remains two issues:

- Sometimes, the output is either empty or consists of garbled text, which we consider as deprecated results (approximately 8% of cases).
- Occasionally, the output triples are incomplete or end with "..." (approximately 3% of cases).

To mitigate these issues, we employ iterated running and re-read prompting strategies to ensure stable outputs. For iterated running, we set the hyperparameter x to control the number of times the entire textual data is re-prompted, with $x = 2$ in this study. For re-read prompting, we set the hyperparameter y to repeatedly feed the same prompt into the model, allowing the LLM to reconsider and refine its answers, with $y = 1$ in this study. These strategies help the LLM generate comprehensive triples from all textual information, with the final output curated through manual checks.

Considering accessibility, we chose to utilize open-source LLMs for the triple generation step, as they are more accessible and convenient to deploy on local servers compared to the GPT series. In this experiment, while `Llama-3-8b` serves as our inference LLM example, other widely-used LLMs like `Llama-2-13b` or fine-tuned models like `PMC-Llama` can also be used in this process.

B. KG Embeddings Comparison

Most current hierarchical KG embeddings are designed to capture and represent hierarchy information in KGs using various geometric structures, including Euclidean and non-Euclidean geometries. Euclidean-based methods typically focus on projections across different coordinates to distinguish entities at various levels. Compared to non-Euclidean approaches like hyperbolic and spherical embeddings, Euclidean projections can efficiently retrieve hierarchical information without additional processes or high computational costs. Additionally, we can separately train embeddings through link prediction tasks, allowing GPU usage to be decoupled from the graph learning process, ensuring stable performance even with more complex KGs. We propose projecting each entity in the KG onto polar coordinates, which enables more efficient embedding retrieval.

Table 2: Different Embedding Baselines Comparison (%)

MODEL	MRR	HITS@1	HITS@3	HITS@10
TransE	54.77	45.90	58.30	72.40
RotatE	57.82	47.95	62.56	75.28
ModE	63.02	55.46	66.25	77.85
TripleRE	64.02	57.21	68.03	79.32
HyperE	59.03	53.69	64.82	74.93
HAKE	**67.37**	**60.23**	**70.62**	**81.21**

Table 2 compares the polar-space-based KG embedding method, which is an adjusted embedding method refered from HAKE (Zhang et al. 2020), with five widely-used baselines (Bordes et al. 2013; Sun et al. 2019; Nickel and Kiela 2017; Fionda and Pirrò 2020), demonstrating the advantages of our approach in retrieving both hierarchical and semantic features from the KG. Performance is evaluated using MRR and HITS@k metrics for regular link prediction tasks, with k set to 1, 3, and 10 in our experiments.

C. Multilevel Attention

After generating $\mathbf{X}$ for all medical codes appears in the intersection of both training set $\mathcal{S}$ and concept vocabulary $\mathcal{C}$, our objective is to utilize EHR dataset to train both the model parameters and the embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{C}| \times d}$ for medical codes, and d is the pre-defined embedding dimension for the previous GNN module. For the training process, we only focus on each patient u with multiple admission records. An encoder is employed to transform all of their admission records into a patient representation vector $\mathbf{p}$, leveraging the embeddings in $\mathbf{X}$:

$$\mathbf{p} = \text{Bi-Attention}(\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_T | \mathbf{X}).$$

In this study, "Bi-Attention" means that we implement a two-tier attention mechanism, detailed as follows:

a) **Learning Admission Embeddings:** Consider an admission record $\mathbf{V}_\tau$ containing n medical codes. The embedding $\mathbf{x}_i$ for each medical code $c_i \in \mathbf{V}_\tau$ can be retrieved using $\mathbf{X}$, so that we can get code embeddings $\mathbf{E}_V^\tau$, which is also the subset of embeddings $\mathbf{E}$, for each admission. To aggregate these code embeddings into an admission embedding $\mathbf{v}_\tau$, we apply a global attention mechanism:

$$\mathbf{E}_V^\tau = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times m}$$

$$\mathbf{z}_i = \tanh(\mathbf{W}_c \mathbf{x}_i) \in \mathbb{R}^a$$

$$\mathbf{v}_\tau = \sum_{i=1}^{n} \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{n} \exp(\mathbf{z}_j)} \mathbf{x}_i \in \mathbb{R}^m.$$

Table 3: Statistics of MIMIC-III and MIMIC-IV datasets

Dataset	MIMIC-III	MIMIC-IV
# patients	7,493	10,000
Max. # visit	42	55
Avg. # visit	2.66	3.66
# codes	4,880	6,102
Max. # codes per visit	39	50
Avg. # codes per visit	13.06	13.38

Here, $\mathbf{W}_c \in \mathbb{R}^{a \times m}$ is the weight matrix, where a represent the attention dimension. The attention scores $\alpha\tau = [\alpha_\tau^1, \alpha_\tau^2, \ldots, \alpha_\tau^n]$ quantify the contribution of each medical code within an admission, and the weighted sum of $\mathbf{x}_i$ results in the admission embedding $\mathbf{v}_\tau$.

b) **Learning Patient Embeddings**: Once the embedding $\mathbf{v}_\tau$ for the τ-th admission is computed, we derive the patient embedding by aggregating across all admissions. The admission embedding $\mathbf{v}_\tau$ is first projected into the patient embedding space through a simple linear classifier:

$$\tilde{\mathbf{v}}_\tau = \sigma(\mathbf{W}_u \mathbf{v}_\tau) \in \mathbb{R}^p.$$

Here, $\mathbf{W}_u \in \mathbb{R}^{p \times m}$ is the corresponding weight matrix, and p denotes the dimension of the patient embedding. The activation function is denoted as σ.

Assume patient u has T admission records as input features of model, we can get the admission embeddings $\mathbf{E}_\mathbf{p}^u$ after the code-level attention module. To aggregate $\mathbf{E}_\mathbf{p}^u$ into the final patient embedding $\mathbf{p}$, we then also apply the global attention across different admissions for each patient:

$$\mathbf{E}_\mathbf{p}^u = [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \ldots, \tilde{\mathbf{v}}_T] \in \mathbb{R}^{T \times p}$$

$$\mathbf{r}_\tau = \tanh(\mathbf{W}_v \tilde{\mathbf{v}}_\tau) \in \mathbb{R}^b$$

$$\mathbf{p} = \sum_{\tau=1}^{T} \frac{\exp(\mathbf{r}_\tau)}{\sum_{\tau=1}^{T} \exp(\mathbf{r}_\tau)} \tilde{\mathbf{v}}_\tau \in \mathbb{R}^p.$$

In this case, $\mathbf{W}_v \in \mathbb{R}^{b \times p}$ is the weight matrix used for the transformation. The attention scores $\beta = [\beta_1, \beta_2, \ldots, \beta_T]$ capture the relative importance of each admission, and their weighted sum produces the final patient embedding $\mathbf{p}$.

D. Experimental Evaluation

D.1 Data Processing Details

To evaluate our proposed model, we focused on two well-established datasets in critical care research: MIMIC-III and MIMIC-IV. Both datasets are derived from de-identified clinical data collected at the Beth Israel Deaconess Medical Center in Boston, Massachusetts, and include detailed records from patients admitted to the Intensive Care Units (ICUs). MIMIC-III contains data from over 40,000 ICU admissions between 2001 and 2012, covering a wide range of information such as patient demographics, vital signs, laboratory results, diagnoses, and diagnostic codes. MIMIC-IV extends this dataset to approximately 60,000 ICU admissions from 2008 to 2019, reflecting more recent clinical practices and patient demographics.

Table 4: Prediction Results on MIMIC-IV for Diagnosis and HF Prediction.

Models	Diagnosis Prediction			HF Prediction	
	w-$\mathbf{F_1}$	R@10	R@20	AUC	$\mathbf{F_1}$
Dipole	$22.16_{(0.2)}$	$36.21_{(0.2)}$	$38.74_{(0.2)}$	$84.80_{(0.3)}$	$69.52_{(0.2)}$
Deepr	$22.58_{(0.1)}$	$36.79_{(0.1)}$	$39.45_{(0.1)}$	$83.61_{(0.5)}$	$70.46_{(0.1)}$
RETAIN	$23.11_{(0.8)}$	$37.32_{(0.8)}$	$40.15_{(0.6)}$	$84.14_{(0.3)}$	$71.23_{(0.2)}$
Timeline	$23.76_{(0.2)}$	$37.89_{(0.1)}$	$40.87_{(0.1)}$	$83.45_{(0.3)}$	$72.30_{(0.2)}$
GRAM	$24.39_{(0.1)}$	$38.42_{(0.2)}$	$41.62_{(0.3)}$	$85.55_{(0.2)}$	$69.82_{(0.4)}$
KAME	$25.01_{(0.1)}$	$38.86_{(0.2)}$	$42.12_{(0.2)}$	$84.80_{(0.2)}$	$72.34_{(0.5)}$
CGL	$25.74_{(0.2)}$	$39.23_{(0.3)}$	$42.67_{(0.2)}$	$87.91_{(0.2)}$	$70.71_{(0.3)}$
G-BERT	$25.12_{(0.3)}$	$39.91_{(0.2)}$	$43.25_{(0.2)}$	$85.76_{(0.2)}$	$72.88_{(0.1)}$
HiTANet	$24.53_{(0.2)}$	$38.42_{(0.3)}$	$41.89_{(0.1)}$	$86.34_{(0.4)}$	$71.35_{(0.2)}$
THAM	$26.97_{(0.3)}$	$43.07_{(0.4)}$	$47.19_{(0.2)}$	$87.43_{(0.3)}$	$72.26_{(0.2)}$
DualMAR	$\mathbf{29.87}_{(0.3)}$	$\mathbf{45.66}_{(0.2)}$	$\mathbf{51.73}_{(0.3)}$	$\mathbf{90.32}_{(0.2)}$	$\mathbf{73.54}_{(0.1)}$

To ensure distinct patient cohorts, we selected patients from MIMIC-IV who were admitted between 2018 and 2019, avoiding temporal overlap with MIMIC-III. We included only patients with multiple visits (# of visits ≥ 2) to ensure sufficient data for analysis. Specifically, for MIMIC-III, we divided the data into 6000 training, 500 validation, and 1000 testing samples. For MIMIC-IV, the distribution comprised 8000 training, 1000 validation, and 2000 testing samples. This approach allows us to leverage the longitudinal structure of EHR data, with the last visit of a patient serving as the label and all preceding visits as features. We aim to enhance the model's ability to accurately predict critical care outcomes, contributing to advancements in predictive modeling and ultimately improving patient care through data-driven insights.

D.2 Parameter Settings

We consider node feature of graph learning module as the hierarchy KG embedding upon polar-space, which is $2k = 2000$ dimension vectors. Specifically, the embedding size $k = 1000$ for both hierarchical and semantic representation in Diagnosis KG. We use two Graph Attention (GAT) Convolution layers for disease-lab graph, where the hidden dimension for both layers are 256, and we remain other settings as default values. The attention sizes $a = 256$ and $b = 256$ remain the same dimension as previous output, and to avoid over-fitting problem, we set dropout layers within both attention layers with 0.2 dropout rate. Moreover, three three-level MLPs are adopted as decoders to help us get lab test embeddings with uniformly 256, 128 hidden dimensions, and we set dropout rate between layers are all 0.4 for either pretraining or direct training process. In addition, to finetune the lab test embeddings for various downstream tasks, a two-level classifiers with 256 hidden dimension and 0.5 dropout rate are used for transforming concatenated features to logits.

To retrieve precise hierarchical and semantic information from Diagnosis KG, we modify the source code of HAKE (Zhang et al. 2020) and set training step as 180,000. For proxy-task learning, we use 10 and 10 epochs for joint and individual training respectively. We uniformly use

Table 5: Wilcoxon Signed-Rank Test and t-Test Results for DualMAR vs. Baseline Models with 10 runs. "p" means p-value for certain evaluation method, and "CI" means Confidence Interval calculated by T-testing method.

Models	Wilcoxon p	T-test p	95% CI
DualMAR vs. Dipole	1.95e-03	1.62e-14	(7.41, 7.80)
DualMAR vs. Deepr	1.95e-03	2.01e-14	(7.17, 7.53)
DualMAR vs. RETAIN	1.95e-03	6.27e-10	(6.46, 7.39)
DualMAR vs. Timeline	1.95e-03	9.77e-13	(5.93, 6.39)
DualMAR vs. GRAM	1.95e-03	5.53e-13	(5.31, 5.67)
DualMAR vs. KAME	1.95e-03	1.23e-12	(4.87, 5.21)
DualMAR vs. CGL	1.95e-03	3.46e-11	(4.49, 4.88)
DualMAR vs. G-BERT	1.95e-03	7.65e-13	(5.62, 6.01)
DualMAR vs. HiTANet	1.95e-03	1.99e-12	(5.89, 6.27)
DualMAR vs. THAM	1.95e-03	4.57e-11	(3.98, 4.32)

Adam optimizer and decay learning rate for all training process, and we use 500 and 50 epochs for diagnosis prediction and heart failure prediction, respectively. Moreover, the initial learning rate in both pretraining and finetuning processes is 0.001. All programs are implemented using Python 3.10 and Pytorch 2.3.1 with CUDA 12.4 on a machine with two AMD EPYC 9254 24-Core Processors , 528GB RAM, and four Nvidia L40S GPUs.

E. Experiment Details

Table 3 shows the basic statistics for MIMIC-III and MIMIC-IV. Table 4 shows the result of main experiment which is the same setting as shown in MIMIC-III. Note that, we remain DualMAR in the same hyperparameter setting as model in MIMIC-III, but it still achieve superior performance across all baselines, which display the effectiveness and robustness of proposed model. Compared to MIMIC-III, MIMIC-IV has larger sample size and more longitude data, which accelerate the predictive performance of DualMAR within more complex scenarios, and we can tune the hyperparameters in model for pursuing even more accurate prediction for various downstream tasks.

Moreover, to evaluate the statistical significance of performance differences between our proposed DualMAR model and baseline models, we employed both the paired t-test and Wilcoxon signed-rank test. The t-test was chosen under the assumption of normally distributed performance metrics. To account for potential deviations from normality, the non-parametric Wilcoxon signed-rank test was also conducted. These tests allow us to robustly determine whether the observed improvements in DualMAR's performance are statistically significant compared to the baselines.

Here we only take the w-F_1 of Diagnosis Prediction in MIMIC-IV as an example to show the evaluation process on Table 5. We observe that the Wilcoxon signed-rank test consistently yielded p-values of 1.95×10^{-3}, indicating significant differences between DualMAR and all baseline models. The paired t-tests further confirmed these results, with all p-values well below 1×10^{-10}. The 95% confidence intervals for the t-tests showed a clear positive difference, reinforcing the superior performance of DualMAR in Diagnosis Prediction compared to each baseline. These results demonstrate the robustness of DualMAR's improvements in weighted F_1 scores over the baselines.

Table 6: Less Frequent Codes Diagnosis Prediction.

Model	Rare-10 R@5	Rare-10 R@8	Rare-10 R@15	Rare-20 R@5	Rare-20 R@8	Rare-20 R@15
CGL	32.48	39.47	52.17	12.53	16.03	24.40
G-BERT	28.97	35.72	48.68	**14.95**	17.11	25.10
HiTANet	21.74	26.09	46.38	11.99	15.43	22.49
THAM	30.43	39.13	43.48	12.95	17.56	23.79
DualMAR	**39.13**	**47.83**	**56.52**	14.28	**18.00**	**26.30**

Table 7: Diagnosis Prediction Results by different KGs.

Models	# Dims	w-F_1	R@10	R@20
DualMAR-$\texttt{KG}_a$	500	$21.75_{(0.3)}$	$33.27_{(0.4)}$	$35.32_{(0.4)}$
DualMAR-$\texttt{KG}_a$	2000	$19.79_{(0.5)}$	$32.12_{(0.4)}$	$33.57_{(0.4)}$
DualMAR-$\texttt{KG}_b$	500	$23.87_{(0.3)}$	$37.73_{(0.4)}$	$39.18_{(0.4)}$
DualMAR-$\texttt{KG}_b$	2000	$24.58_{(0.3)}$	$39.27_{(0.4)}$	$40.32_{(0.4)}$
DualMAR-$\texttt{KG}_{a+c}$	500	$23.56_{(0.2)}$	$37.43_{(0.3)}$	$38.74_{(0.4)}$
DualMAR-$\texttt{KG}_{a+c}$	2000	$23.47_{(0.3)}$	$37.89_{(0.4)}$	$38.54_{(0.4)}$
DualMAR-$\texttt{KG}_{b+c}$	500	$24.32_{(0.3)}$	$39.73_{(0.4)}$	$40.15_{(0.4)}$
DualMAR-$\texttt{KG}_{b+c}$	2000	$\mathbf{25.37}_{\mathbf{(0.3)}}$	$\mathbf{40.52}_{\mathbf{(0.2)}}$	$\mathbf{41.86}_{\mathbf{(0.3)}}$

F. More Case Studies

We design experiments focusing on predicting less frequent condition codes in MIMIC-III, similar to rare ICD code prediction tasks. We conduct two subtasks:

- `Rare-20`: Predicting 200 ICD codes that occur fewer than 20 times in MIMIC-III.
- `Rare-10`: Predicting 100 ICD codes that occur fewer than 10 times in MIMIC-III.

Table 6 shows the full results of both `Rare-10` and `Rare-20`. It highlights DualMAR's superior performance in predicting less frequent ICD codes across both subtasks, `Rare-10` and `Rare-20`. Specifically, DualMAR consistently outperforms baseline models, achieving the highest Recall at all thresholds (R@5, R@8, R@15). Notably, for `Rare-20`, DualMAR significantly surpasses other models, particularly in R@8 and R@15, reflecting its strong capability in handling rare condition codes. These results underscore the effectiveness of DualMAR in scenarios where capturing rare but critical diagnoses is crucial.

Table 7 shows the full results of DualMAR with varying KGs and embedding sizes. Beyond analysis for w-F_1 in the case study, full table also demonstrates the Recall performance of DualMAR. The results indicate that incorporating KG components ($\texttt{KG}_{\texttt{b+c}}$) consistently yields the highest Recall across R@10 and R@20, particularly with 2000-dimensional embeddings. Notably, $\texttt{KG}_{\texttt{b}}$ also shows strong performance, suggesting the significant contributions of fundamental Ontology-KG in enhancing diagnostic predictions. The comparison highlights the critical role of augmenting KG to achieve superior recall in diagnosis prediction tasks.

References

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2787–2795.

Brennan, M. D.; Miner, K. M.; and Rizza, R. A. 1998. The Mayo Clinic. *The Journal of Clinical Endocrinology & Metabolism*, 83(10): 3427–3434.

Fionda, V.; and Pirrò, G. 2020. Learning Triple Embeddings from Knowledge Graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 3874–3881. AAAI Press.

Gargano, M. A.; Matentzoglu, N.; Coleman, B.; Addo-Lartey, E. B.; Anagnostopoulos, A. V.; Anderton, J.; Avillach, P.; Bagley, A. M.; Bakštein, E.; Balhoff, J. P.; et al. 2024. The Human Phenotype Ontology in 2024: phenotypes around the world. *Nucleic acids research*, 52(D1): D1333–D1346.

Knox, C.; Wilson, M.; Klinger, C. M.; Franklin, M.; Oler, E.; Wilson, A.; Pon, A.; Cox, J.; Chin, N. E.; Strawbridge, S. A.; et al. 2024. DrugBank 6.0: the DrugBank knowledgebase for 2024. *Nucleic acids research*, 52(D1): D1265–D1275.

Kuhn, M.; Letunic, I.; Jensen, L. J.; and Bork, P. 2016. The SIDER database of drugs and side effects. *Nucleic acids research*, 44(D1): D1075–D1079.

Nickel, M.; and Kiela, D. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6338–6347.

Organization, W. H.; et al. 1988. International classification of diseases—Ninth revision (ICD-9). *Weekly Epidemiological Record= Relevé épidémiologique hebdomadaire*, 63(45): 343–344.

Schieppati, A.; Henter, J.-I.; Daina, E.; and Aperia, A. 2008. Why rare diseases are an important medical and social issue. *The Lancet*, 371(9629): 2039–2041.

Sun, Z.; Deng, Z.; Nie, J.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Ursu, O.; Holmes, J.; Knockel, J.; Bologa, C. G.; Yang, J. J.; Mathias, S. L.; Nelson, S. J.; and Oprea, T. I. 2016. DrugCentral: online drug compendium. *Nucleic acids research*, gkw993.

Weinreich, S. S.; Mangon, R.; Sikkens, J.; Teeuw, M. E.; and Cornel, M. 2008. Orphanet: a European database for rare diseases. *Nederlands tijdschrift voor geneeskunde*, 152(9): 518–519.

Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 3065–3072.