Medication Recommendation based on Diffusion Knowledge Graphs

Yang Zhao

December 2024

Abstract

Medical Recommendation Systems (MRS) have become essential in healthcare, leveraging advancements in Large Language Models (LLMs) and Knowledge Graphs (KGs). Many efforts have been investigated in this field, however, two primary challenges hinder the development of reliable Medical Expert Systems (MES): insufficient domain knowledge on complete and noisy data limited generative recommendation. To overcome these issues, I applied a modified diffusion model that integrates a noising-denoising process on a patient-diagnosis-drug KG. I reconstruct the patient-diagnosis-medication graph using the MIMIC-3 dataset and map it to the Drug-Drug Interaction (DDI) KG using consistent encodings. The combined KGs are input into the diffusion model to predict patient-medication connections probabilistically. A collaborative filtering module further refines personalized recommendations. Lastrly, I compare the proposed diffusion-based method with existing graph-based approaches and validate its improved accuracy through experiments and case studies.

1 Introduction

MRS have gained significant popularity due to the rapid development of LLMs and KGs. A notable example is the IBM Watson Health program [9], which collaborates with Google Health, Microsoft Health, and Nvidia Health. This program aims to provide powerful diagnostic analysis as well as accurate and efficient medication recommendations, leveraging its extensive backend knowledge dataset and powerful online, objective-oriented computation engines. The research challenges summarized from their experience and other teams' findings highlight that insufficient domain knowledge and excessive irrelevant noise significantly hinder the further investigation of MES. These challenges need to be urgently explored and addressed.

Specifically, the first challenge, insufficient medication knowledge, refers to the gap between clinical data and biomedical KGs, such as Electronic Health

Records (EHR) data [28]. Although the patient-diagnosis and drug-diagnosis relationships have been widely studied recently, connecting the patient-diagnosis-drug relationship remains challenging. This difficulty arises due to subject barriers among different encoding standards and the complex molecular and chemical interactions at different stages of disease progression.

The second difficulty is more closely related to the recommendation system itself. Noisy data, including inconsistencies between user interactions and underlying medical knowledge, can negatively affect the accuracy and reliability of recommendations. While implicit feedback, such as recorded prescriptions or treatment outcomes, is often used to understand patient needs, it can be misleading [15]. For instance, a prescribed medication may not fully address a patient's condition due to incomplete diagnosis or external factors, making it harder to establish accurate patient-diagnosis-medication relationships [3]. This noise can confuse recommendation models and reduce their effectiveness in suggesting the most suitable treatments. [4]

To address the aforementioned problems, I integrated the noising and denoising process on a processed patient-diagnosis-drug KG to train a diffusion model by incorporating the loss of the MRS recommendation task between a patient's diagnosis and medications. Specifically, I rebuilt the patient-diagnosis-medication KG from the MIMIC-3 dataset and mapped it to the DDI KG using the same encoding methods. The combined historical MIMIC-3 and DDI KGs are then fed into a modified diffusion model to output patient-medication connections with different probabilities. Finally, the recommended medication ranks are extracted from the last collaborative filtering module, which considers personalized recommendation tasks. Additionally, I compared the proposed modified diffusion-based recommendation method with other knowledge graph-based and graph-based methods on overall evaluation performance and specific case study.

2 Related Works

2.1 Medication Knowledge-aware Recommendation Systems

The method of MRS could be summarized the following key approaches: Knowledge Graph-Based Methods, Collaborative Filtering Approaches, Deep Learning Models, and Hybrid Methods. Knowledge graphs have become a cornerstone in many MRS frameworks, as they provide structured representations of entities and relationships. For example, [28] proposed PharmKG, which integrates biomedical knowledge graphs with patient-specific data to enhance drug recommendations. Similarly, DDI graphs [18] are often utilized to identify potential conflicts or synergies between medications, improving the safety and effectiveness of recommendations. Collaborative filtering methods have been widely ap-

plied in recommendation systems, including MRS. These methods typically rely on historical patient-medication interaction data to predict future prescriptions. Techniques such as matrix factorization and deep collaborative filtering [8] have demonstrated significant potential in identifying personalized medication recommendations. However, these approaches often face challenges in dealing with sparse or noisy data, limiting their effectiveness in real-world applications. Deep learning models, particularly those using sequence-based architectures, have also been explored for MRS. Recurrent Neural Networks and Transformers are commonly used to model temporal dependencies in EHR data. For instance, [6] developed RETAIN, an interpretable model that uses attention mechanisms to recommend treatments based on patient histories. Furthermore, generative models like Variational Autoencoders (VAEs) [14] and diffusion-based models [22] have recently been introduced to better capture complex relationships between patient conditions and medications. Hybrid methods that combine multiple techniques, such as KGs, collaborative filtering, and deep learning, have shown promise in overcoming individual limitations. For example, hybrid models often use knowledge graphs to address data sparsity issues in collaborative filtering and employ deep learning to capture intricate patterns in patient-medication interactions [1].

2.2 Sparse Knowledge Graph Recommendation System

Sparse KG recommendation systems aim to leverage knowledge graph representations to enhance recommendation tasks in scenarios where user-item interaction data is sparse. The integration of KGs into recommendation models has shown promise in addressing the cold-start problem, improving explainability, and leveraging additional contextual knowledge. Knowledge Graph Embedding methods is used to capture semantic relationships within sparse KGs for recommendation. Early works such as TransE [5] and TransR [27] encode KG entities and relations into low-dimensional vector spaces. These embeddings have been integrated with collaborative filtering techniques to enhance recommendation performance in sparse data scenarios. Models such as CrossRec [20] and CoNet [10] leverage auxiliary domain KGs to augment sparse target KGs, sharing knowledge across domains. Additionally, context-aware recommendation systems [2] incorporate user-specific contextual information into KG-based models to refine sparse interaction data and improve recommendations.

2.3 Generative Recommendation System

Generative Recommendation Systems have gained significant attention for their ability to model user preferences and item interactions by learning data distributions. Unlike traditional methods, which primarily rely on historical interactions and embeddings, generative approaches aim to synthesize recommendations through learned generative processes. MultiVAE [14] introduced a variational autoencoder with multinomial likelihood to capture implicit feedback

in user-item interaction data. Extensions such as RecVAE [17] improved upon MultiVAE by incorporating regularization techniques to address overfitting and enhance recommendation performance. These methods excel in capturing user-item interaction distributions, especially in sparse datasets. GANs have been applied to recommendation systems for generating realistic user-item interaction data. DiffRec [22] employs a diffusion generative model to infer user preferences by modeling the probabilities of user-item interactions, demonstrating state-of-the-art performance in sparse recommendation settings. Large Language Models (LLMs) such as GPT and BERT have shown potential in recommendation systems by generating recommendations as a sequence prediction task. Models like GPT-Rec [11] utilize transformer-based architectures to capture sequential dependencies in user interactions, generating personalized recommendations. These models excel in scenarios involving sequential or session-based recommendation tasks. Hybrid approaches combine generative models with other recommendation techniques to leverage their respective strengths.

3 Method

The method on generative medical recommendation worked on diffusion knowledge graph focus on parts: efficient fusion knowledge aggregated from surrounding entities, diffusion process to output as much as varied relations among triplets, and filtered reliable recommendations feedback. So, I here primarily applied the method which proposed in DiffKG [12] to integrate medication KGs on it. And the figure 1 is the pipeline of this method under my understanding. The following subsections just shown the primary part of the whole method.

3.1 Entity Knowledge Aggregation

The first part is about the comprehensive embedding of entities in KG. To sufficient consider the potential positive relation between entities, like the different drugs with specific molecular structures might be able to medicate to other diagnosis, the aggregations from neighboring entities are fully investigated and learned. Specifically, for knowledge graph G_K, we have:

$$\mathbf{x}_i = \text{Drop}\left(\text{Norm}\left(\mathbf{x}_i + \sum_{e \in \mathcal{N}_i} \alpha(e, r_{e,i}, i)\mathbf{x}_e\right)\right),$$

$$\alpha(e, r_{e,i}, i) = \frac{\exp\left(\text{LeakyReLU}\left(r_{e,i}^\top W[\mathbf{x}_e \| \mathbf{x}_i]\right)\right)}{\sum_{e \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(r_{e,i}^\top W[\mathbf{x}_e \| \mathbf{x}_i]\right)\right)},$$

where, $\mathcal{N}_i$: the neighboring entities of item i. $\mathbf{x}_i \in \mathbb{R}^d$: embedding of item. Drop: dropout operation to prevent overfitting. $\alpha(e, r_{e,i}, i)$: attentive relevance during knowledge aggregation process, to capture distinct semantics of relationships between i and e. $r_{e,i}$: relation type. $\mathbf{x}_e \in \mathbb{R}^d$: embedding of entity. Norm: normalization operation. $W \in \mathbb{R}^{d \times 2d}$: weight matrix to customize the input i and e.LeakyReLU: non-linear activation function.

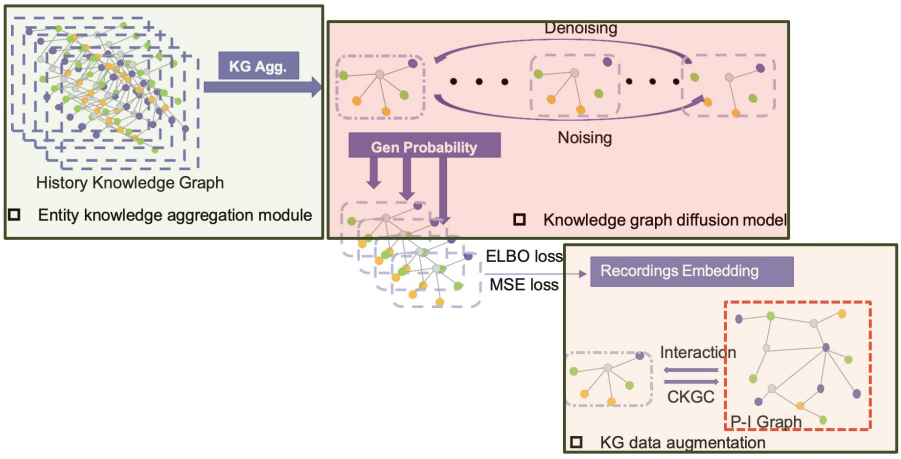

Figure 1: Pipeline of Medication Diffusion Recommendation System: There are three main modules: the Entity Knowledge Aggregation Module, the Knowledge Graph Diffusion Module, and the KG Data Augmentation Module. The integrated KG, which combines historical data from MIMIC-3 and DDI, is fed into the Entity Knowledge Aggregation Module in the form of a binary patient-diagnosis-drug graph (D-D). In this graph, a value of 1 denotes that a diagnosis p can be treated with drug D, while a value of 0 indicates otherwise. The Entity Knowledge Aggregation Module mitigates the sparsity of diagnosis D by aggregating information from its neighboring diagnoses and medication procedures. Next, the aggregated embedding entities are passed to the Knowledge Graph Diffusion Module. In this module, Gaussian noise is added to the entities during the forward process. The reverse process denoises the entities by optimizing the similarity between the original entities' distribution and the generated distribution. Finally, the KG Data Augmentation Module applies a collaborative filtering layer to refine the recommendations. This layer maximizes the agreement among positive pairs (e.g., related patients and drugs) while minimizing the agreement among negative pairs (e.g., unrelated patients and drugs) within the patient-drug (P-D) graph. This process is performed simultaneously to improve the recommendation accuracy and reliability.

3.2 Optimizing the diffusion generation and

The aggregated triplets are fed to diffusion model by adding Gaussain noise and reversing by removing noising according to the learnable way (typically a neural network). By optimizing the evidence lower bound objective on similarity between the ground-truth and inferred data distribution at different noise level. This loss could be denoted as $\mathcal{L}_{\text{elbo}}$.

Further, in order to aggregate the patient diagnosis-medication interaction and DDI data into denoised KG to enhance its relevance to recommendation tasks, applyong the Collaborative Knowledge Graph Convolution layer (CKGC)

before the final output layer of diffusion model. Thus, the training objective which optimizing ELBO and CKGC loss simultaneously is transformed as following:

$$\mathcal{L}_{\text{kgdm}} = (1 - \lambda_0)\mathcal{L}_{\text{elbo}} + \lambda_0 \mathcal{L}_{\text{ckgc}}$$

where, the MSE loss to CKGC could be defined as:

$$\mathcal{L}_{\text{ckgc}} = \left\| \left[\mathcal{A} \cdot \hat{\mathbf{x}}_0^\top \right]^\top \cdot \mathbf{E}_p - \mathbf{E}_i \right\|_2^2$$

3.3 Generated KG reconstruction

Reconstruct G'_k from G_k, only containing the relationships relevant to the downstream recommendation tasks. For the Graph Embedding Propagation Layer with $\mathbf{x}_p^{(l+1)} = \sum_{i \in \mathcal{N}_p} \frac{\mathbf{x}_i^{(l)}}{\sqrt{|\mathcal{N}_p| \cdot |\mathcal{N}_i|}}$, and $\mathbf{x}_i^{(l+1)} = \sum_{p \in \mathcal{N}_i} \frac{\mathbf{x}_p^{(l)}}{\sqrt{|\mathcal{N}_i| \cdot |\mathcal{N}_p|}}$, the Graph-based collaborative filtering (CF) captures collaborative signals of higher order by maximizing the agreement among positive pairs and minimize the agreement among negative pairs.

$$\mathcal{L}_{\text{cl}}^{\text{patient}} = \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\mathbf{x}'_u, \mathbf{x}''_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\mathbf{x}'_u, \mathbf{x}''_v)/\tau)}$$

$$\mathcal{L}_{\text{cl}} = \mathcal{L}_{\text{cl}}^{\text{patient}} + \mathcal{L}_{\text{cl}}^{\text{item}}$$

where, $\mathbf{x}_i^{(l)}$: the encoded representations of item i. $\mathbf{x}_u^{(l)}$: the encoded representations of user u. $\mathcal{N}_i$: the neighboring entities of item i. $\mathcal{N}_u$: the neighboring entities of item u. $s(\cdot)$: cosine similarity. $(\mathbf{x}'_u, \mathbf{x}''_v) \mid u, v \in \mathcal{U}, u \neq v$: negative pairs (the different node pairs). $(\mathbf{x}'_u, \mathbf{x}''_u) \mid u \in \mathcal{U}$: positive pairs (the same node pairs). $\mathcal{L}_{\text{cl}}^{\text{user}}$: contrastive loss of user, $\mathcal{L}_{\text{cl}}^{\text{item}}$: contrastive loss of item.

So, the Overall Loss of DiffKG could be defined as optimizing recommendation task by Bayesian personalized ranking (BPR):

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{bpr}} + \lambda_1 \mathcal{L}_{\text{cl}} + \lambda_2 \|\Theta\|_2^2,$$

where, $\mathcal{L}_{\text{bpr}} = \sum_{(u,i,j) \in \mathcal{O}} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj})$. $\lambda_0, \lambda_1, \lambda_2$: hyperparameters of strength respectively. $\mathcal{O}^+$: observed interaction from the Cartesian product of user and item set. $\mathcal{O}^-$: observed interaction from the Cartesian product of user and item set. Θ: learnable parameters set of the model.

Algorithm 1: Pipeline for Knowledge Graph Modification

Original KG relations x_0, items, entities, k Updated Knowledge Graph G'_k

Procedure `AddNoiseToKG(`x_0`):`
    Apply noise to x_0;
    return $x_{T'}$;

Procedure `ReverseDenoise(`$x_{T'}$`):`
    Initialize $\hat{x}_T = x_{T'}$;
    for $t = T$ **to** 1 **do**
        $\hat{x}_{t-1} = \mu_\theta(\hat{x}_t, t)$ `// Deterministic denoising`;
    return $\hat{x}_T$;

Procedure `ReconstructKG(`$\hat{x}_T$`):`
    Use $\hat{x}_T$ to build G'_k;
    return G'_k;

Procedure `AddTopKRelations(`G'_k*, items, entities,* k`):`
    foreach *item* i *in items* **do**
        Select top-k entities $J = \{j_1, j_2, ..., j_k\}$ based on scores;
        foreach *entity* $j \in J$ **do**
            Add relation between item i and entity j in G'_k;
    return G'_k;

Function *MainPipeline*x_0*, items, entities,* k**:**
    $x_{T'} \leftarrow$ `AddNoiseToKG(`x_0`)`;
    $\hat{x}_T \leftarrow$ `ReverseDenoise(`$x_{T'}$`)`;
    $G'_k \leftarrow$ `ReconstructKG(`$\hat{x}_T$`)`;
    $G'_k \leftarrow$ `AddTopKRelations(`G'_k*, items, entities,* k`)`;
    return G'_k;

4 Experiment

To evaluate the effectiveness of the DiffKG framework applied to our prepared MIMIC-3 and DDI KG for medication recommendation, we have designed a series of experiments to address the following research questions:

- **RQ1:** How does the performance of the DiffKG framework compare against various state-of-the-art recommendation systems in the context of medication recommendation tasks?
- **RQ2:** How effectively does the DiffKG framework enhance the interpretability of its recommendations, providing deeper insights into its decision-making process for medication recommendations?

4.1 Data

The experiments are carried out on MIMIC-3 [13]. The dataset consists of EHR collected over 11 years, with each record corresponding to one visit. These records include diagnosis, procedure, and medication information. The diagnosis and procedure codes are further merged according to the ICD-9 standard. 2 shows the typical examples of MIMIC-3 KG with three patients and their corresponding medical procedure, diagnosis, and medication index encoded by ATC.

```
User_3 <has_procedure> Procedure_[1, 0, 13, 22, 27, 31, 14, 39, 28, 41, 26, 32, 17, 29]
User_4 <has_diagnosis> Diagnosis_[63, 64, 65, 66, 67, 68, 20, 47, 46, 52, 69, 70]
User_4 <has_diagnosis> Diagnosis_[28, 29, 30, 31, 32, 33, 2, 15]
User_4 <has_diagnosis> Diagnosis_[0, 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 13, 2, 22, 26, 40, 41, 28, 42, 43, 44, 37, 45, 46, 47, 18, 20, 48, 49, 50, 17, 51, 52]
User_4 <has_procedure> Procedure_[71, 65, 72, 73, 74, 75, 76, 68, 46, 52, 77, 70]
User_4 <has_procedure> Procedure_[34, 28, 35, 36, 37, 38, 39]
User_4 <has_procedure> Procedure_[1, 5, 6, 0, 13, 22, 2, 29, 26, 38, 28, 41, 32, 45, 18, 17, 52, 44, 12, 39, 11, 42, 3, 47, 53]
User_4 <has_medication> Medication_[20, 50, 68, 46, 78, 69, 52, 70, 79]
User_4 <has_medication> Medication_[21, 40, 41, 14, 42, 43, 4, 1]
User_4 <has_medication> Medication_[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 22, 26, 40, 29, 47, 18, 48, 17, 51, 39, 16, 53, 37, 54]
User_5 <has_diagnosis> Diagnosis_[80, 81, 82, 12, 83, 84, 85, 86, 87]
```

Figure 2: Screenshot of MIMIC-3 knowledge graph

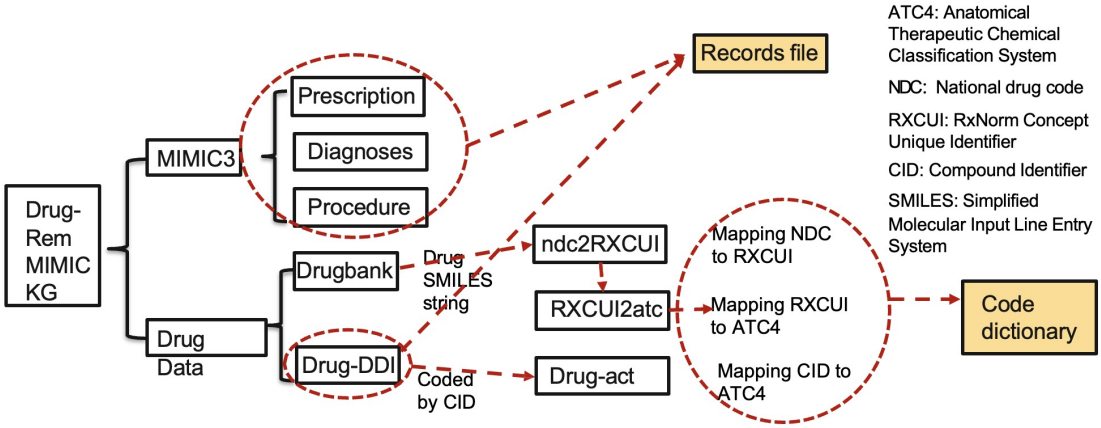

Figure 3: Relationship among Datasets

```
(0, 0, 1)   (0, 3000, 1)   (0, 6043, 1)   (1, 3007, 1)
(0, 1, 1)   (0, 3001, 1)   (0, 6044, 1)   (1, 3009, 1)
(0, 2, 1)   (0, 3002, 1)   (0, 6046, 1)   (1, 3010, 1)
(0, 3, 1)   (0, 3003, 1)   (0, 6048, 1)   (1, 3012, 1)
(0, 6, 1)   (0, 3006, 1)   (0, 6049, 1)   (1, 3013, 1)
(0, 7, 1)   (0, 3007, 1)
(0, 8, 1)   (0, 3012, 1)
            (0, 3017, 1)
```

Figure 4: Screenshot of flatten MIMIC-3 + DDI knowledge graph without duplicates

For DDI information, the top-40 severity types are obtained from TWOSIDES [19], and the drugs are subsequently coded according to the same ATC level. Drug molecular information is extracted from SMILES strings obtained from [7], and interactions are derived using *RDKit's Chem*.

The EHR data, along with the DDI data, are processed following the methodology proposed by [26]. The process begins with a longitudinal patient representation module that learns patient representations from their EHR data. These patient representations are then passed to a global message-passing neural network (MPNN) encoder, which outputs a global drug vector. Each entry in this vector quantifies the similarity between the patient representation and each drug representation. Next, a bipartite encoder, also using the patient representation as input, generates a local drug vector that encodes the molecular substructure functionalities of the drugs. Finally, the global and local drug vectors are element-wise combined in the medication representation module, where the final drug representation is obtained through element-wise thresholding. The relations among used dataset is shown in 3.

Lastly, the triplets extracted from the above process was further flatten and processed to only keep the unique entities in 4. and relations with scope [0, 2999] for diagnosis, scope[3000, 5999] for procedure, and scope [6000, 8999] for medication. The statistics of the post-processed data are reported in Table 1.

Table 1: Dataset Statistics for MIMIC3-Drug

Statistics	**MIMIC3-Drug**
# Patients	6,350
# Diagnoses	1,917
# Procedures	1,898
# Medication	1,593
# Interacion	40,326
# Density	1.28×10^{-5}
Knowledge Graph	
# Entities	3,612
# Relations	4
# Triplets	41,000

4.2 Evaluation

Recall@N and NDCG@N (Normalized Discounted Cumulative Gain) are commonly used evaluation metrics in information retrieval, recommendation systems, and ranking tasks. They could be defined as following:

$$\text{Recall@N} = \frac{\text{Number of relevant items in the top-N results}}{\text{Total number of relevant items}} \tag{1}$$

Where, N denotes the number of top-ranked items considered. Number of relevant items in the top-N results represents the count of relevant items present

within the top-N retrieved results. Total number of relevant items denotes the total count of relevant items available for the query or dataset.

$$\text{DCG@N} = \sum_{i=1}^{N} \frac{\text{rel}_i}{\log_2(i+1)} \tag{2}$$

Where, rel_i: is the relevance score of the item at position i. And i is the rank position of the item in the retrieved list.

$$\text{IDCG@N} = \sum_{i=1}^{N} \frac{\text{rel}_i^{\text{ideal}}}{\log_2(i+1)} \tag{3}$$

Where $\text{rel}_i^{\text{ideal}}$ represents the relevance score of the item in the ideal ranking.

NDCG@N is obtained by normalizing the DCG by the IDCG:

$$\text{NDCG@N} = \frac{\text{DCG@N}}{\text{IDCG@N}} \tag{4}$$

Where, rel_i represents the relevance score of the item at position i. $\text{rel}_i^{\text{ideal}}$ is the relevance score of the item in the ideal ranking. N is the number of top-ranked items considered. Recall@N focuses only on the presence of relevant items in the top-N results, without considering their order. NDCG@N takes into account both the relevance and the ranking position, giving higher weight to relevant items appearing earlier in the ranking. Both of the two evaluation represents better when values are bigger.

4.3 Baselines

For a comprehensive evaluation, we thoroughly compare our DiffKG with a diverse set of baselines derived from different research streams.

GNN-based KG-enhanced Recommenders .

- **KGCN** [21]: The implementation of this method is based on their released GitHub repository: `https://github.com/hwwang55/KGCN.git`.
- **KGAT** [23]: The implementation of this method is based on their released GitHub repository: `https://github.com/xiangwang1223/knowledge_graph_attention_network.git`.
- **KGIN** [24]: The implementation of this method is based on their released GitHub repository: `https://github.com/huangtinglin/Knowledge_Graph_based_Intent_Network.git`.

Generatio-based KG-enhanced Recommenders .

- **MultiVAE** [14]: The implementation of this method is based on their released GitHub repository: `https://github.com/dawenl/vae_cf.git`.
- **CDAE** [25]: The implementation of this method is based on their released GitHub repository: `https://github.com/henry0312/CDAE.git`.
- **DiffRec** [22]: The implementation of this method is based on their released GitHub repository: `https://github.com/YiyanXu/DiffRec.git`.

4.4 Hyper-parameters

For all baseline models, I directly applied their default or recommended hyper-parameters from their original experiments. For DiffKG, the parameters from the original paper vary significantly across datasets such as Last-FM, MIND, and Alibaba-iFashion, which have different structures compared to our dataset, MIMIC-3 + DDI. Specifically, differences include the User-Item (U-I) graph settings and graph sparsity. To adapt to these differences, I began by testing the recommended hyper-parameters from the original paper and then made further modifications and adjustments. The final experimental settings are summarized in Table 2.

Table 2: Hyper-parameters for DiffKG

Category	**Hyper-parameter**	**Value / Tuning Range**
Diffusion Setting	Diffusion steps (T)	50
	Noise lower bound ($\alpha_{\min}$)	1×10^{-4}
	Noise upper bound ($\alpha_{\max}$)	1×10^{-3}
	Noise scale (s)	$\{1 \times 10^{-4}, 1 \times 10^{-3}\}$
	Denoising Neural Network	MLP
Loss-related	Loss balance factor (λ)	$\{0.1, 0.2, \ldots, 0.6\}$
	Denoising weight factor (σ)	$\{0, 0.05, 0.1, 0.2, \ldots, 0.9\}$

5 Results

5.1 Overall Results

The experimental results, as shown in Table 3, demonstrate that the DiffKG model outperforms other baseline models across multiple key metrics for medical recommendation tasks, utilizing the MIMIC-3 and DDI knowledge graph. DiffKG effectively leverages the structural and relational information from the knowledge graph to address challenges like data sparsity and enhance its ability to recommend medications.

Notably, DiffKG achieves the highest Recall@20 (0.0716) and excels in ranking-based metrics such as NDCG@20 (0.2515) and NDCG@10 (0.4378), highlighting its superior ability to prioritize and rank relevant recommendations. These strengths make it well-suited for medical applications requiring accurate and reliable medication suggestions.

While DiffKG slightly lags behind CDAE in Recall@10, this indicates potential for further improvement, particularly for smaller recommendation sets. Future enhancements, such as integrating user-specific noise handling or collaborative filtering techniques, could improve its performance in this area. Overall, DiffKG demonstrates state-of-the-art performance, particularly excelling in ranking relevance, making it a robust framework for precise and clinically relevant medical recommendations.

Table 3: Performance comparison of models on recommendation metrics.

Model	**Recall@20**	**Recall@10**	**NDCG@20**	**NDCG@10**
DiffKG (ours)	**0.0716**	0.1300	**0.2515**	**0.4378**
MultiVAE	0.0693	0.1210	0.1988	0.3757
CDAE	0.0701	**0.1331**	0.1917	0.3642
DiffRec	0.0706	0.1292	0.2471	0.4223
KGIN	0.0615	0.1239	0.2234	0.3843
KGCN	0.0621	0.1292	0.1999	0.3718
KGAT	0.0572	0.1196	0.1764	0.3681

5.2 Case Study

To further analyze the performance of DiffKG, a case study was conducted to compare its medication recommendations against other baseline methods, including MultiVAE, CDAE, and KGIN. The case study is based on two diagnosis scenarios, with the corresponding recommendations summarized in Table 4.

The case study highlights DiffKG's ability to provide broader and more targeted medication recommendations compared to other methods. For both diagnosis scenarios, DiffKG suggests a diverse and clinically meaningful set of medications, effectively balancing specificity and comprehensiveness. For instance, DiffKG includes additional medications such as *Amlodipine* and *Nifedipine*, demonstrating its strength in addressing hypertension-related conditions and complex comorbidities. These results reflect DiffKG's capability to leverage the underlying knowledge graph structure to model intricate relationships between diagnoses and medications.

Table 4: Diagnosis and medicine recommendations for different methods.

Diagnosis	**Methods**	**Medicine Recommendations**
Sepsis, Acute respiratory failure, Hypertension	MultiVAE	Metoprolol Tartrate, Vancomycin, Furosemide
	CDAE	Furosemide, Metoprolol, Insulin, Norepinephrine
	KGIN	Vancomycin, Metoprolol Tartrate, Corticosteroids
	DiffKG	Furosemide, Amlodipine, Norepinephrine, Acetaminophen, Corticosteroids
Type 2 diabetes, Rheumatoid arthritis, Hypertension, Hyperlipidemia	MultiVAE	Phenylbutazone, Insulin, Fenofibrate, Empagliflozin, Liraglutide
	CDAE	Metformin, Tolbutamide, Phenylbutazone, Insulin, Acetaminophen, Empagliflozin, Liraglutide
	KGIN	Metformin, Amethopterin, Amiloride/HCTZ, Fenofibrate, Empagliflozin, Liraglutide
	DiffKG	Metformin, Insulin, Acetaminophen, Nifedipine, Fenofibrate

6 Conclusion and Future Work

In this work, we evaluated the DiffKG framework for medication recommendation tasks using the MIMIC-3 and DDI knowledge graph. The results demonstrate that DiffKG outperforms baseline models in key metrics such as Recall and NDCG, highlighting its ability to provide clinically relevant and well-ranked medication recommendations. This superior performance underscores the effectiveness of leveraging knowledge graph structures and diffusion-based mechanisms to address challenges like data sparsity and noise. Furthermore, the case study illustrates DiffKG's capacity to generate diverse and targeted recommendations by modeling complex relationships between diagnoses and medications.

While the overall performance comparison confirms DiffKG's strong capability in medication recommendation tasks, further investigations are necessary. Specifically, hyper-parameter tuning and key module ablation studies should be conducted to enhance DiffKG's adaptability and performance in real-world medical systems. Additionally, the datasets used to evaluate DiffKG's performance are derived from historical knowledge graphs. Thus, its effectiveness in online medication recommendation systems, where real-world patient-drug feedback introduces more complex noise, remains to be thoroughly explored.

Future work will focus on integrating DiffKG with SSLRec [16] to collaborate with other modules and improve its generative capabilities in real-world scenarios. This integration aims to address the unique challenges posed by dynamic and noisy environments, ensuring the framework's scalability and reliability in practical medical applications.

References

[1] Omolola A Adeoye-Olatunde, Olga O Vlashyn, Kimberly S Illingworth Plake, Jamie L Woodyard, Zachary A Weber, Debra K Litzelman, and Alissa L Russ-Jara. A mixed-methods study of pharmacy instructors' early experiences with a teaching electronic medical record. *Currents in Pharmacy Teaching and Learning*, 13(9):1180–1193, 2021.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2010.

[3] Yaser Al-Worafi. *Patient safety in developing countries: Education, research, case studies*. CRC Press, 2023.

[4] Paolo Biancone, Silvana Secinaro, Roberto Marseglia, and Davide Calandra. E-health for the future. managerial perspectives using a multiple case study approach. *Technovation*, 120:102406, 2023.

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[6] 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, 2016.

[7] DrugBank. Drugbank online. `https://go.drugbank.com/`. Accessed: December 12, 2024.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[9] IBM Waston Health. Ibm waston health. `https://www.ibm.com/industries/healthcare`. Accessed: December 12, 2024.

[10] Guangneng Hu, Yu Zhang, and Qiang Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 667–676, 2018.

[11] Jiaxiong Hu, Junze Li, Yuhang Zeng, Dongjie Yang, Danxuan Liang, Helen Meng, and Xiaojuan Ma. Designing scaffolding strategies for conversational agents in dialog task of neurocognitive disorders screening. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024.

[12] Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. Diffkg: Knowledge graph diffusion model for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 313–321, 2024.

[13] 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.

[14] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[15] Kannan Nova. Generative ai in healthcare: advancements in electronic health records, facilitating medical languages, and personalized patient care. *Journal of Advanced Analytics in Healthcare Management*, 7(1):115–131, 2023.

[16] Xubin Ren, Lianghao Xia, Yuhao Yang, Wei Wei, Tianle Wang, Xuheng Cai, and Chao Huang. Sslrec: A self-supervised learning framework for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 567–575, 2024.

[17] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining*, pages 528–536, 2020.

[18] Nicholas P Tatonetti, Guy Haskin Fernald, and Russ B Altman. A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports. *Journal of the American Medical Informatics Association*, 19(1):79–85, 2012.

[19] Nicholas P Tatonetti, Patrick P Ye, Roxana Daneshjou, and Russ B Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.

[20] Hanfei Wang, Yuan Zuo, Hong Li, and Junjie Wu. Cross-domain recommendation with user personality. *Knowledge-Based Systems*, 213:106664, 2021.

[21] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313, 2019.

[22] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 832–841, 2023.

[23] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[24] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, pages 878–887, 2021.

[25] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*, pages 153–162, 2016.

[26] Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. Safedrug: Dual molecular graph encoders for recommending effective and safe drug combinations. *arXiv preprint arXiv:2105.02711*, 2021.

[27] Hongxi Zhao, Hongfei Li, Qiaoming Liu, Guanghui Dong, Chang Hou, Yang Li, and Yuming Zhao. Using transr to enhance drug repurposing knowledge graph for covid-19 and its complications. *Methods*, 221:82–90, 2024.

[28] Shuangjia Zheng, Jiahua Rao, Ying Song, Jixian Zhang, Xianglu Xiao, Evandro Fei Fang, Yuedong Yang, and Zhangming Niu. Pharmkg: a dedicated knowledge graph benchmark for bomedical data mining. *Briefings in bioinformatics*, 22(4):bbaa344, 2021.