Disentangled Dynamic Heterogeneous Graph Learning for Opioid Overdose Prediction

Qianlong Wen
qwen@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Zhongyu Ouyang
zouyang2@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Jianfei Zhang
jianfei.zhang@live.ca
University of Alberta
Edmonton, Alberta, Canada

Yiyue Qian
yqian5@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Yanfang Ye*
yye7@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Chuxu Zhang*
chuxuzhang@brandeis.edu
Brandeis University
Waltham, Massachusetts, USA

ABSTRACT

Opioids (e.g., oxycodone and morphine) are highly addictive prescription (aka Rx) drugs which can be easily overprescribed and lead to opioid overdose. Recently, the opioid epidemic is increasingly serious across the US as its related deaths have risen at alarming rates. To combat the deadly opioid epidemic, a state-run prescription drug monitoring program (PDMP) has been established to alleviate the drug over-prescribing problem in the US. Although PDMP provides a detailed prescription history related to opioids, it is still not enough to prevent opioid overdose because it cannot predict over-prescribing risk. In addition, existing machine learning-based methods mainly focus on drug doses while ignoring other prescribing patterns behind patients' historical records, thus resulting in suboptimal performance. To this end, we propose a novel model DDHGNN - **D**isentangled **D**ynamic **H**eterogeneous **G**raph **N**eural **N**etwork, for over-prescribing prediction. Specifically, we abstract the PDMP data into a dynamic heterogeneous graph which comprehensively depicts the prescribing and dispensing (P&D) relationships. Then, we design a dynamic heterogeneous graph neural network to learn patients' representations. Furthermore, we devise an adversarial disentangler to learn a disentangled representation which is particularly related to the prescribing patterns. Extensive experiments on a 1-year anonymous PDMP data demonstrate that DDHGNN outperforms state-of-the-art methods, revealing its promising future in preventing opioid overdose.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; • **Theory of computation → Dynamic graph algorithms**; • **Computing methodologies → Neural networks**.

KEYWORDS

Dynamic Heterogeneous Graph, Graph Neural Network, Opioid Overdose, PDMP

ACM Reference Format:
Qianlong Wen, Zhongyu Ouyang, Jianfei Zhang, Yiyue Qian, Yanfang Ye, and Chuxu Zhang. 2022. Disentangled Dynamic Heterogeneous Graph Learning for Opioid Overdose Prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539279

1 INTRODUCTION

Opioids are commonly used for pain relief among prescription (aka Rx) drugs, while the intense pleasure they bring to patients can easily lead to addiction and even overdose. Although the legal dispensation of opioids requires medical prescription from licensed physicians, the opioid-related deaths in the US have still gone through an alarming increase. According to the report from CDC, the number rose from 21,088 in 2010 to 46,802 in 2018 [16]. Early prediction and intervention of over-prescribing behaviors could be a key to alleviate such a problem. However, relying on the evaluation of professional healthcare workers is not sufficient in many cases. Fortunately, increasingly advanced machine learning techniques enable us to detect/predict potential over-prescribing patients on large-scale datasets. Traditional machine learning methods, like regression, gradient boost and random forest model, have been employed to estimate opioid overdose risk [15, 19]. More recently, some deep learning methods have also been proposed for opioid overdose prediction. Depending on the data they use, they model dynamic dependency [2, 9], spatial dependency [8, 39], or combine the two dependencies together [5, 38] to solve the problem. Despite the progress that has been made, few of the previous methods have explored the prescribing patterns of over-prescribing patients, which could help us predict potential overdose patients early, since these patterns are time-evolving.

In the United States, a state-run prescription drug monitoring program (PDMP) [4] is established to collect and distribute data about the prescription and dispensation of federally controlled substances and other potentially addictive prescription drugs from pharmacies in the form of electronic databases. However, the utilization of PDMP among professional healthcare workers is still limited and one of the major reasons is that the PDMP dataset is unable to find potential over-prescribing patients [13]. Thus, an

[*] Corresponding authors.

KDD '22, August 14–18, 2022, Washington, DC, USA.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539279

effective model which can find patients at risk based on their historical prescription records is needed. To achieve this goal, some challenges need to be addressed: (1) Due to the feature distribution difference among the Rx entries in PDMP data (e.g., patients and drugs) and the relations between them (e.g., patient-drug relation and patient-physician relation), capturing the prescribing and dispensing (P&D) relationships among different Rx entries is the first challenge; (2) The relations among different Rx entries are naturally dynamic and the intervals between them varies. Therefore, modeling the spatial and dynamic dependencies is the second challenge; (3) Rx drugs can be prescribed and dispensed by different physicians and pharmacies, and prescribed with various doses with repeated refills. These factors could have different impacts on patients. Thus, extracting the informative factors behind the PDMP data is the third challenge.

To address the above challenges, we propose a novel **D**isentangled **D**ynamic **H**eterogeneous **G**raph **N**eural **N**etwork (DDHGNN) to predict potential patients with high over-prescribing risks. Specifically, we first construct a dynamic heterogeneous graph to abstract the PDMP data and properly describes the P&D relations among different Rx entries. To handle the spatial and dynamic dependencies simultaneously while preserving the heterogeneity, we propose a dynamic heterogeneous graph attention network (DHGAT) enhanced with a functional time encoding strategy. Although the entry representations learned by DHGAT can be directly applied to predict potential over-prescribing patients, multiple factors are highly entangled in the representations, which makes the task more challenging. Thus, with patient embedding obtained from DHGAT, we further design a novel disentangler based on generative adversarial network to extract the factors specific to the prescribing patterns. In particular, we introduce external prior knowledge generated from different relation views and a prior-exchange mechanism to make the disentangled representation more reliable. Finally, the disentangled patient embeddings are used for predicting the potential high-risk over-prescribing patients. To summarize, our contributions in this work are:

- To combat the increasingly serious opioid overdose problem, we propose to predict patients with high overdose risk based on the PDMP data. We abstract the P&D reltions in PDMP data into a dynamic heterogeneous graph to not only properly describe relationships between different Rx entries but also integrate spatial and dynamic dependencies simultaneously.
- A novel model called DDHGNN with two collaborative components (i.e., DHGAT and prior-enhanced adversarial disentangler) is proposed to encode spatial and dynamic properties on a dynamic heterogeneous graph and further separate informative factors from the learned embeddings.
- Extensive experiments are conducted on a 1-year PDMP dataset. The proposed DDHGNN achieves state-of-the-art performance by comparison with many baseline methods, demonstrating its effectiveness and promising future in preventing opioid epidemic.

2 RELATED WORK

2.1 Heterogeneous Graph Learning

Due to the heterogeneous relations in PDMP data, our work is related to heterogeneous graph embedding (HGE) . Generally, there are three kinds of HGE methods [35]: (1) The proximity-preserving HGE methods [3, 37] which are mostly based on random walk [25] and optimized by skip-gram [12]; (2) The message-passing HGE methods that consider both of structural and node attribute information. These methods usually learn graph embedding by aggregating and transforming the embeddings of the original neighbors [18, 36, 40] or meta-path neighbors [10, 30, 41]; (3) The relation-learning HGE methods [1, 31, 34] which transform the heterogeneous networks into schema-rich knowledge graph and optimize the embeddings by measuring the acceptability of the unseen triplets. On the other line, embedding learning on dynamic graphs has also drawn increasing attention. These methods can be roughly divided into two categories: (1) Discrete-time methods [7, 24, 27] which discretize a dynamic graph into a sequence snapshots and encode them with static methods to produce a series of embeddings, then fit the embeddings into time series models. Therefore, the temporal information is lost within the same snapshot, which is unsuitable for real-world data with continuous temporal information; (2) Continuous-time methods [21, 23, 26, 32], which directly operate on dynamic graph without time discretization. To handle the dynamics, different temporal encoding techniques are raised. However, the aforementioned methods mainly target homogeneous graphs and their performance might be limited to heterogeneous graphs. Although a number of models for dynamic heterogeneous graphs have also been proposed [6, 33], their performance still either suffer from the time discretization strategy or node attribute information loss. In this work, we propose to build a dynamic heterogeneous graph model which handles the dynamics continuously and preserves heterogeneity.

2.2 Over-prescribing Prediction

In this work, we aim to make early predictions of patients within the PDMP system who have high risks in overdose. There have been some studies for opioid overdose prediction or detection. We roughly assort previous methods into three categories according to different data characteristics that they tend to model: (1) Temporal methods [9, 15, 19] model time series data, like medication or diagnosed results, to predict individual or regional opioid overdose risk as time goes; (2) Spatial methods [8, 39] detect opioid users from relational data, such as social network. With the detection results, extra caution on these opioid users will prevent worse situations happening; (3) Spatio-temporal methods [5, 38] combine both dynamic and spatial dependencies together to capture more complex relations, generally leading to better performances. Despite the great successes of existing studies on over-prescribing prediction, most of them are not able to make timely predictions to effectively prevent opioid-overdose because the opioid addiction is not directly associated with the variables they use for prediction. In light of this, we further exploit PDMP data and utilize the dynamic heterogeneous relations among different Rx entries to capture the prescribing patterns which directly reflect opioid addiction.

3 PRELIMINARIES

In this section, we will first introduce some concepts used throughout this paper, then formally define the problem.

Definition 3.1. **Heterogeneous Graph.** A heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathrm{X}_{\mathcal{V}}, \mathrm{X}_{\mathcal{E}})$ consist of a node set $\mathcal{V}$, an edge set $\mathcal{E}$ along with the node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$, and the edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{A}$ and $\mathcal{R}$ denotes the node and edge types, with the constrain $|\mathcal{A}| + |\mathcal{R}| > 2$. In addition, each node $v \in \mathcal{V}$ (and edge $(u, v) \in \mathcal{E}$) could be associated with a attribute vector $\mathbf{x}_v \in \mathbf{X}_{\mathcal{V}}$ (and $\mathbf{x}_{(u,v)} \in \mathbf{X}_{\mathcal{E}}$).

Definition 3.2. **dynamic heterogeneous Graph.** In this work, a dynamic heterogeneous graph is constructed by a sequence of chronologically ordered events $\mathcal{G}_T = \{e(t_1), e(t_2), \ldots\}$ and multiple events could occur at the same time ($0 \leq t_1 \leq t_2 \leq \ldots \leq T$). Each event $e(t)$ can be represented by a quadruplet (u, v, r, t), which indicates an interaction between node u and node v with relation r at time t. $\mathcal{V}_T = \{v : \forall v \in n(t), t \in [0, T]\}$ is the node set, $\mathcal{E}_T = \{(u, v) : \forall (u, v) \in n(t), t \in [0, T]\}$ represents the edge set, $\mathcal{R}_T = \{r : \forall r \in n(t), t \in [0, T]\}$ denotes the relation type set, respectively. Any node has never been seen before will be added to the graph first and it would be a multi-graph if there are more than one event between a pair of nodes.

The prescriptions in a temporal order connected by the heterogeneous relations among different Rx entries (i.e., patient, drug, physician and pharmacy) makes the constructed **dynamic P&D graph** naturally dynamic and heterogeneous. Take the prescription in Figure 1(a) as an example, a 35-year-old male patient visits a psychiatrist physician who prescribes him 21 tablets of a Rx drug (e.g., Tramadol) with 200mg/tablet for a 7-day supply. The patient fills the prescription on 15/03/2016 at a pharmacy which is permitted to dispense such Rx drugs. The corresponding graph schema is shown in Figure 1(b), where five types of P&D relationships are used to describe the interactions among four types of Rx entries. Each refill behavior is a time-stamped event and the corresponding nodes and edges associated with each record are added to the graph accordingly. For a patient and his/her prescription records, the process to construct a dynamic P&D graph G_t is shown in Figure 1(c). A pair of nodes might interact with each other more than once.

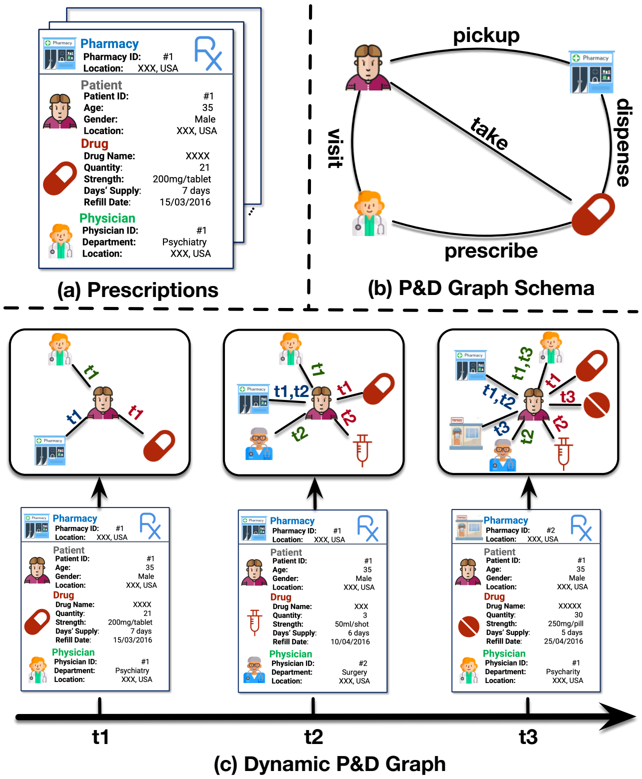
Definition 3.3. **Opioid Over-prescribing Prediction.** Given a constructed dynamic P&D graph $\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T, \mathbf{X}_{\mathcal{V}_T}, \mathbf{X}_{\mathcal{E}_T})$ up to time T, the task is to develop a machine learning model to learn representation $\mathbf{h}_v$ of each patient v, which is further utilized to predict opioid overdose risk of v after timestamp T.

4 MODEL

In this section, we present details of the proposed model DDHGNN, which is shown in Figure 2. It is composed of two key components: 1) Dynamic Heterogeneous graph attention network (DHGAT), which includes intra-relation temporal aggregation and inter-relation aggregation mechanisms, to learn relation-specific embeddings and comprehensively fuse them together, respectively; and 2) Prior-enhanced adversarial disentangler, which encodes and decodes the patient embeddings obtained from the previous step to generate more informative disentangled embeddings. The pseudo-code of DDHGNN is shown in Appendix A, we elaborate these two components in the following of this section.

4.1 Patient Embedding with Dynamic Heterogeneous Graph Attention Network

To generate patient embeddings, we develop a dynamic heterogeneous graph attention network which includes two consecutive steps: first use an intra-relation temporal aggregation mechanism to summarize temporal messages from neighbors connected by each relation, and then apply an inter-relation aggregation mechanism to combine previous relation-specific embeddings.



(a) Prescriptions
(b) P&D Graph Schema
(c) Dynamic P&D Graph

Figure 1: (a) An example of a prescription; (b) P&D graph schema; (c) Example of dynamic P&D graph construction based on a patient's prescription records.

4.1.1 Intra-relation Temporal Aggregation. We first introduce an intra-relation temporal aggregation mechanism, which fuses neighbor information within the same relation type. In a heterogeneous graph, each node and edge type may have its own attribute space. Taking the PDMP dataset as an example, the initial patient attributes are associated with their demographic information (e.g., age and sex), while the initial drug attributes reflect its chemical components and side effects. To address the attributes' heterogeneity, we employ a type-specific projection on each node and edge to map their distinct raw attribute vectors into the same feature space. Specifically, given the attribute vector $\mathbf{x}_v \in \mathbb{R}^{d_{\phi(v)}}$ of node v and $\mathbf{x}_{(u,v)} \in \mathbb{R}^{d_{\psi(u,v)}}$ of edge (u, v), their projections are formulated as:

$$\begin{aligned} \mathbf{h}_v^0 &= \sigma\left(\mathbf{W}_{\phi(v)} \cdot \mathbf{x}_v\right), \\ \mathbf{h}_{(u,v)}^0 &= \sigma\left(\mathbf{W}_{\psi(u,v)} \cdot \mathbf{x}_{(u,v)}\right), \end{aligned} \tag{1}$$

where $\mathbf{h}_v^0, \mathbf{h}_{(u,v)}^0 \in \mathbb{R}^d$; $\mathbf{W}_{\phi(v)} \in \mathbb{R}^{d \times d_{\phi(v)}}$ and $\mathbf{W}_{\psi(u,v)} \in \mathbb{R}^{d \times d_{\psi(u,v)}}$ are the trainable type-specific projection parameters for node type $\phi(v)$ and edge type $\psi(u, v)$, respectively; $\sigma(\cdot)$ is *ReLU* function. For each node, neighbors connected by the same relation can contribute differently to its embedding (e.g., drug-patient impact varies with drug type, dose, and refill time). In light of this, we introduce a

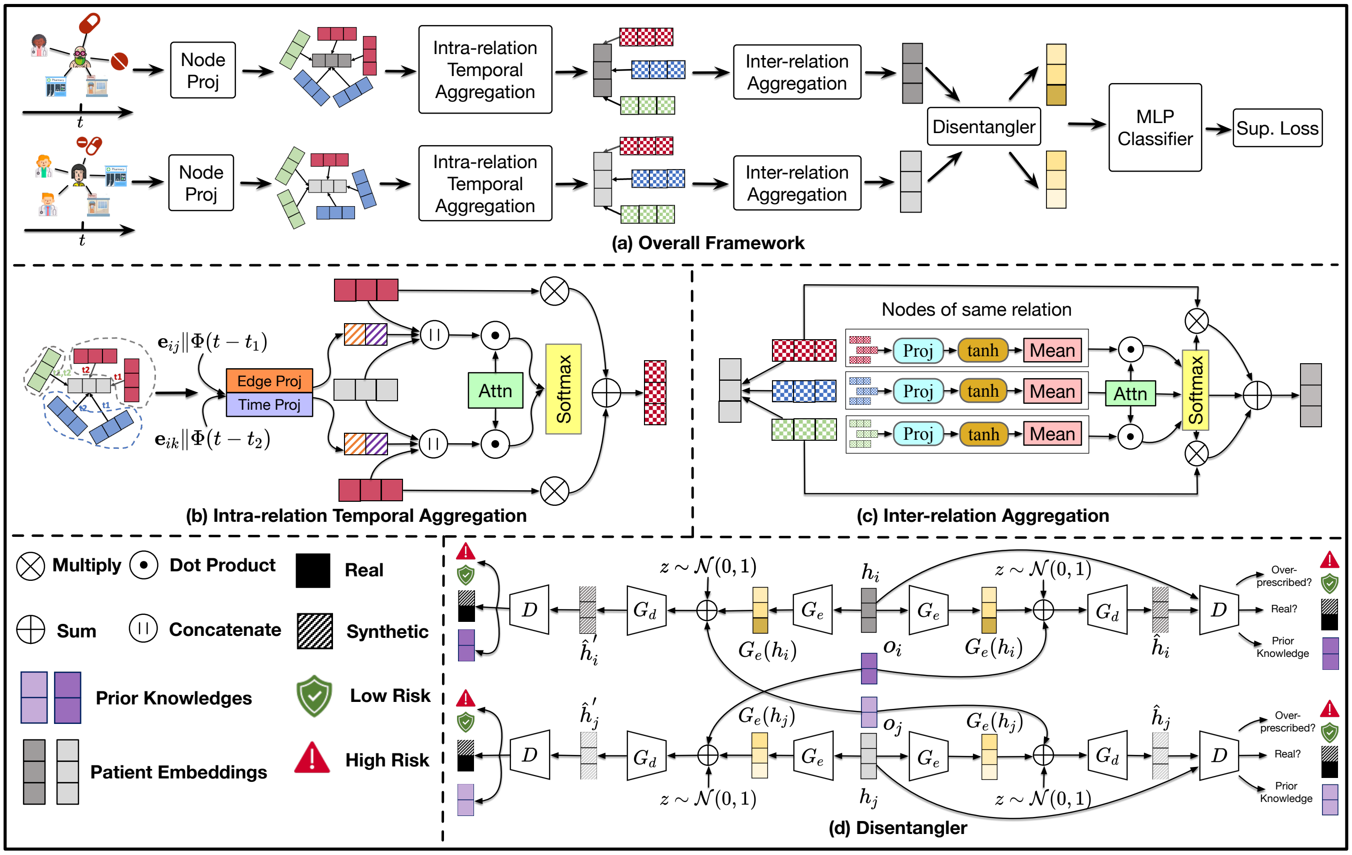


Figure 2: (a) The overall framework of DDHGNN. Raw features are first projected to the embedding space and are later used for intra- and inter-relations aggregations. Embeddings are then refined by disentangling from prior knowledge and are passed into an MLP for further prediction; (b) For each relation, combine edge and time projections to compute relation attention coefficients, which are later used to aggregate temporal relation-specific messages; (c) Aggregate relation-specific messages with the attention mechanism; (d) Prior-enhanced adversarial disentangler balances the dosage and behavior patterns by training the GAN with cross-labeled paired samples.

time-aware self-attention mechanism to learn the weight for each neighbor of the same type. Particularly, given a target node v and its relation-r-based neighbors $\mathcal{N}^r_T(v)$ at timestamp T, attention score of node $u \in \mathcal{N}^r_T(v)$ is calculated through three steps. For each edge, we first concatenate its source node embedding, target node embedding, edge embedding, and time encoding together as the message vector. Then, we multiply the message vector of each edge with an attention vector to generate its attention score. Finally, we normalize the attention score across all relation-r-based neighbors through $Softmax$ operation. Without loss of generality, we extend it to multi-head mechanism which is formulated as:

$$\mathbf{m}^l_{(u,v),T} = \left[\mathbf{h}^{l-1}_{u,T} \| \mathbf{h}^{l-1}_{v,T} \| \mathbf{h}^0_{(u,v),T} \| \Phi(T - t_{(u,v)})\right],$$
$$\boldsymbol{\alpha}^{l,r}_{(u,v),T} = \overset{K}{\underset{k=1}{\|}} \sigma\left(\underset{\forall u\in\mathcal{N}^r_T(v)}{\mathrm{Softmax}}\left(\left[\mathbf{a}^{l,r}_T\right]_k \cdot \left[\mathbf{m}^l_{(u,v),T}\right]_k\right)\right), \tag{2}$$

where $\|$ denotes the concatenation operation, K is the number of heads, $\Phi(\cdot) \in \mathbb{R}^d$ is a generic time encoding function [32], $t_{(u,v)}$ is the timestamp that edge (u,v) appears, $\sigma(\cdot)$ is $LeakyReLU$ function, $\mathbf{a}^{l,r}_T \in \mathbb{R}^{4d}$ are the trainable attention vectors. Given the computed attention scores of node v's neighbors and their embeddings in $(l-1)$ layer, we perform weighted aggregation to get the embedding of node v specific to relation r:

$$\mathbf{h}^{l,r}_{v,T} = \sigma\left(\sum_{u\in\mathcal{N}^r_T(v)} \boldsymbol{\alpha}^{l,r}_{(u,v),T} \cdot \left[\mathbf{W}^{l,r}_{M,T} \cdot \mathbf{h}^{l-1}_{u,T}\right]\right), \tag{3}$$

where $\mathbf{W}^{l,r}_{M,T} \in \mathbb{R}^{d\times d}$ is message transformation matrix. Figure 2(c) shows the illustration of intra-relation temporal aggregation.

4.1.2 Inter-relation Aggregation. Through intra-relation aggregation, we gather a set of relation-specific embeddings for node v, denoted as $\left\{\mathbf{h}^{l,r_1}_{v,T}, \mathbf{h}^{l,r_2}_{v,T}, \ldots, \mathbf{h}^{l,r_m}_{v,T}\right\}$ (m is the number of relations associated with node v). Then, we fuse the multiple relation-specific node embeddings to learn comprehensive node embeddings. Instead of taking the element-wise average of these relation-specific embeddings, we propose to use the inter-relation aggregation mechanism (Figure 2(b)) to automatically learn the relation-specific importance. For a specific relation r, we apply a non-linear transformation to it and summarize it by averaging:

$$\mathbf{s}_T^{l,r} = \frac{1}{\left|\mathcal{V}_T^r\right|} \sum_{v' \in \mathcal{V}_T^r} \left[\tanh\left(\mathbf{W}_T^{l,R} \cdot \mathbf{h}_{v',T}^{l,r} + \mathbf{b}_T^{l,R}\right)\right], \tag{4}$$

where $\mathcal{V}_T^r$ is the set nodes connected by relation r at timestamp T; $\mathbf{W}_T^{l,R} \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_T^{l,R} \in \mathbb{R}^d$ are the trainable transformation weight and bias, respectively. We then multiply the summarized embeddings with a relation attention vector to compute the relations-specific attention scores, followed by a $Softmax$ operation:

$$\beta_{v,T}^{l,r} = \underset{\forall r \in \mathcal{R}(v)}{\text{Softmax}} \left(\mathbf{c}_T^{l,R} \cdot \mathbf{s}_T^{l,r}\right), \tag{5}$$

where $\mathbf{c}_T^{l,R} \in \mathbb{R}^d$ is the trainable relation attention vector. Finally, we aggregate the relation-specific embeddings with the normalized relation-specific attention score:

$$\widetilde{\mathbf{h}}_{v,T}^{l} = \sum_{r \in R(v)} \beta_{v,T}^{l,r} \cdot \mathbf{h}_{v,T}^{l,r}. \tag{6}$$

Meanwhile, a gate mechanism is utilized to control how much information the aggregation result should contribute. Given the aggregation result of node v in layer l and the output of layer $l-1$, we combine them as:

$$\mathbf{h}_{v,T}^{l} = Norm\left(\delta_{\phi(v)} \cdot \widetilde{\mathbf{h}}_{v,T}^{l} + \left(1 - \delta_{\phi(v)}\right) \cdot \mathbf{h}_{v,T}^{l-1}\right), \tag{7}$$

where $Norm(\cdot)$ denotes the layer-norm function, $\delta_{\phi(v)} \in \mathbb{R}$ is the trainable residual weight. A single DHGAT layer includes a intra-relation aggregation module and a inter-relation module. By stacking L DHGAT layers, we derive the patient embedding $\mathbf{h}_{v,T}^{L}$, equally denoted as $\mathbf{h}_v$ to be the input of the later prior-enhanced adversarial disentangler To this end, we denote Θ as the set of parameters in the two modules.

4.2 Prior-Enhanced Adversarial Disentangler

Although DHGAT proposed in the previous subsection is effective in integrating both dynamic and spatial dependencies over the dynamic P&D graph, some factors residing in the generated embeddings might be redundant for our overprescribing prediction task. These factors are unable to depict the dynamic trend behind patients' prescription records (e.g., patients with a large number of prescriptions can still be considered as low risk if they refill their prescriptions regularly at a safe level). This issue further calls for more informative factors in describing patients' prescribing patterns. In particular, we consider three kinds of anomalous behaviors: (1) patients visit multiple physicians for the same kind of Rx drugs; (2) patients pick up the same kind of Rx drugs at different pharmacies, especially the not-so-close ones; (3) patients use multiple kinds of Rx drugs simultaneously, which could indirectly lead to overdose even when the dose level of each single Rx drug is not dangerous. With these patterns slowly evolving with time, the informative factors and other static information could be highly entangled with each other, resulting in suboptimal performance. To alleviate the issue, we would like to learn disentangled patient embeddings which only consist of factors specific to their prescribing patterns. Between the two patterns, the behavior pattern is harder to extract compared with the dosage pattern since the anomalous behaviors are not quantified in the raw data. Therefore, we propose to generate anomalous behavior-independent priors from three relation-based perspectives (i.e., patient-physician, patient-pharmacy and patient-drug) denoted as $\mathbf{o}_i \in \mathbb{R}^{1\times 3}$ to make learning behavior pattern factors more reliable [22]. Computation examples of the priors are illustrated in Appendix C.

To incorporate the generated prior knowledge in deriving the disentangled patient embeddings, we propose a novel prior-enhanced adversarial disentangler (Figure 2(d)), which consists of a generator G and a discriminator D. The discriminator $D = \left[D^T, D^L, D^C\right]$ is a multi-task neural network consisting of three parts: (1) D^T competes with G and distinguishes a real patient embedding from a synthetic one; (2) D^L performs the over-prescribing prediction; (3) and D^C aims to recover the priors from the input embeddings. The generator G combats with D by generating synthetic embeddings $\widehat{\mathbf{h}_i}$ through an encoder-decoder framework. Specifically, the generator $G = [G_e, G_d]$, where the encoder G_e learns to map the original patient embeddings $\mathbf{h}_i$ to a disentangled representation $G_e(\mathbf{h}_i)$, while the decoder G_d incorporates $G_e(\mathbf{h}_i)$ with node i's prior knowledge $\mathbf{o}_i$ and Gaussian noise $\mathbf{z} \sim \mathcal{N}(0,1)$ to generate the synthetic embedding $\widehat{\mathbf{h}_i}$. This process is formulated as:

$$\hat{\mathbf{h}_i} = G_d\left(G_e(\mathbf{h}_i), \mathbf{o}_i, \mathbf{z}\right). \tag{8}$$

Meanwhile, to balance the learning between dosage pattern and behavior pattern, we exchange the behavior-independent prior of each patient and then follow the same process to generate another synthetic embedding (the left part of Figure 2(d)). Additionally, we feed the patient embedding to the disentangler with a cross-labeled pair (one labeled as high-risk of overdosing in the future and the one as low-risk) and exchange the priors within this pair. By doing so, we enforce the disentangler to make a correct prediction even without his/her behavior pattern information, thus learn better dosage pattern factors. With the adversarial disentangler designed in a "drop and recover" manner, we expect G_e to learn a mapping from the original embedding to the disentangled representation where prior knowledge is excluded. Formally, the objective function of D is formulated as:

$$\begin{aligned} \max_D V_D(D,G) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{h})}\left[\log D^T(\mathbf{h}) - D^L(\mathbf{h}) - D^C(\mathbf{h})\right] + \\ \mathbb{E}_{p_{\mathrm{m}}(\mathbf{h})}\left[\log\left(1 - D^T(\hat{\mathbf{h}})\right) - D^L(\hat{\mathbf{h}}) - D^C(\hat{\mathbf{h}})\right] + \\ \mathbb{E}_{p_{\mathrm{e}}(\mathbf{h})}\left[\log\left(1 - D^T\left(\hat{\mathbf{h}}'\right)\right) - D^L\left(\hat{\mathbf{h}}'\right) - D^C\left(\hat{\mathbf{h}}'\right)\right], \end{aligned} \tag{9}$$

where we use cross-entropy loss for D^T and D^L, and mean square error for D^C. These three parts (i.e., $\mathbb{E}_{p_{\mathrm{d}}}$, $\mathbb{E}_{p_{\mathrm{m}}}$ and $\mathbb{E}_{p_{\mathrm{e}}}$) correspond to the aforementioned three tasks on the real inputs, synthetic inputs recovered with their own priors and exchanged priors, respectively. It is noteworthy that D^C will recover the prior knowledge of another patient in the paired samples in the third part of Eq. (9). Similarly, we formulate the objective function of G as:

$$\begin{aligned} \max_G V_G(D,G) = & \mathbb{E}_{p_{\mathrm{d}}(\mathbf{h})}\left[\log D^T(\hat{\mathbf{h}}) - D^L(\hat{\mathbf{h}}) - D^C(\hat{\mathbf{h}})\right] + \\ & \mathbb{E}_{p_{\mathrm{d}}(\mathbf{h})}\left[\log D^T(\hat{\mathbf{h}'}) - D^L(\hat{\mathbf{h}'}) - D^C(\hat{\mathbf{h}'})\right]. \end{aligned} \tag{10}$$

4.3 Objective Function

With the proposed two components, we learn a disentangled embedding $G_e(\mathbf{h}_v)$ for each patient v, which is further utilized for opioid over-prescribing prediction. Specifically, we first feed the disentangled embedding into a multi-layer perceptron (MLP) to predict this patient's over-prescribing label:

$$\widehat{\boldsymbol{y}}_v = \mathrm{MLP}\left(G_e(\mathbf{h}_v)\right), \tag{11}$$

where $\widehat{\boldsymbol{y}}_v$ stands for the predicted overdose probability. Then, we average cross-entropy losses over all labeled patients $\mathcal{V}_{patient}$ as the final objective:

$$L_p = -\sum_{v \in \mathcal{V}_{patient}} \boldsymbol{y}_v^T \log \widehat{\boldsymbol{y}}_v, \tag{12}$$

where $\boldsymbol{y}_v$ is the ground truth label of patient v. Then, we combine L_p with the objective of the disentangler (i.e., $V_G(D,G)$ and $V_D(D,G)$) to formulate the final objective function for model training:

$$\min_{\Theta,D,G} \left(L_p - V_G(D,G) - V_D(D,G)\right). \tag{13}$$

5 EXPERIMENTS

In this section, we first introduce our collected PDMP data, and then conduct extensive experiments to comprehensively evaluate the proposed model.

5.1 Dataset

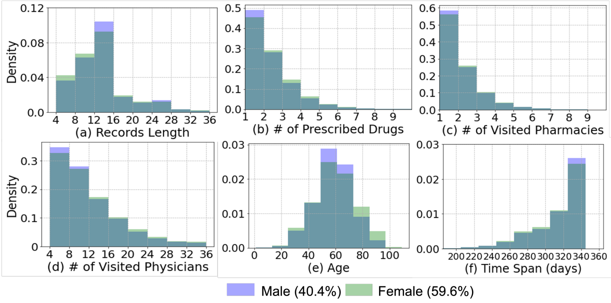


Figure 3: Distributions of different data statistics.

The experimental dataset is collected from the PDMP of Ohio State spanning the year of 2016, which consists of patients' medical prescribing and dispensing records throughout the year. Each record contains basic profiles of the corresponding patient (e.g., age and sex), pharmacy and physician (e.g., specialty), the prescribing and dispensing dates, as well as the information related to the dispensed drug (e.g., drug class, supply days, and prescribed dosage). In total, 4,791,522 records of 366,990 patients, 2,747 pharmacies, 38,122 physicians, and 17 opioids in combination with different specifications are sampled for experiments. The selected opioids are listed in Appendix B. Figure 3 shows the distributions of patients' record length, number of prescribed drugs in combination with their specifications, number of visited pharmacies and physicians, as well as those of patients' age and records' time span in days. By comparison, statistical distributions for male and female groups are of minor difference and record-associated attributes (e.g., record length, time span) are either left or right skewed.

5.2 Experimental Setups

Data Preparation. We refer to the Morphine Milligram Equivalent (MME) metric [11] stressed by CDC as the standard to label the data. Specifically, we transform the opioid daily dose of patients to MME based on their prescription records and then assign patients with positive labels if their daily MME exceed the CDC recommended cut-off (i.e., 90 MME/day) for a few days, otherwise negative. The details of MME and data labeling are provided in Appendix D. In our experiments, we utilize the first 9, 10 or 11 months' records to predict labels regarding the future 3, 2 or 1 month(s), creating three different data splits. For example, given the previous 9 months' prescribing history, we aim to predict whether a patient is prone to be overprescribed in the next 3 months. Note that for each *history-future* split variant, none of the patients are labeled positive given only their historical records because our goal is predicting patients at future risks rather identifying patients' current statuses. Therefore, each split variant has a different number of positive and negative patients and their statistics are provided in Appendix D. Furthermore, we trial two *train-val-test* split ratios: train/val/test = 60%/10%/30% and 70%/10%/20%. In total, there are 3 × 2 set of experimental settings. Moreover, because PDMP data is intensively imbalanced (i.e., most of the patients have low overdose risk), the majority group (patients with negative labels) is down-sampled to construct a 50/50 balanced dataset (positive group with equal size of negative group). We list the size of positive group, negative group and downsampled negative group in Appendix D.

Baselines. To evaluate the performance of our proposed model, we select 10 baseline models to compare with DDHGNN, including one temporal sequence model (i.e., LSTM [17]), three static GNN models (i.e., GCN [20], GAT [29] and GraphSage [14]), as well as six dynamic GNN models (i.e., HGT [18], CTDNE [23], TGAT [32], TGN [26], JODIE [21] and RxNet [38]), where HGT and RxNet are originally designed for heterogeneous graphs. We explain the implementation details of these baselines in Appendix E.

Reproducibility. We implement our model with Python (3.8.5), Pytorch (1.9.1) and DGL (0.7.1). We use a three-layer-MLP as the prediction layer in Eq. (11) and employ Adam and OneCycleLR in Pytoch as the optimizer and scheduler. The code is included in the supplement file and the dataset will be available upon paper publication. For baselines, we adopt LSTM's open-source implementation in Pytorch and borrow GCN, GAT, GraphSage and HGT's open-source implementations in DGL. For CTDNE, TGAT, TGN, JODIE and RxNet, we refer the source codes provided by their authors. We carefully tune the hyper-parameters of DDHGNN and baseline models by grid search to obtain the best performance. More details regarding hyper-parameter settings can be found in Appendix F.

Evaluation Metrics. Accuracy, F1 score and Area under the ROC Curve (ROC-AUC) are utilized to evaluate the performance of different models for overprescribing risk prediction. Each experiment is repeated three times with different random seeds. The average results with variance are reported.

5.3 Overall Performance Comparison

The prediction performance of all models are shown in Table 1, from which we have the following observations: (1) Dynamic models (including LSTM and dynamic GNN models) significantly outperform static models (i.e., GCN, GAT, GraphSage) on all the experimental settings. This phenomenon is most likely caused by the loss of temporal information in P&D graph. Thus, the static model cannot identify the more recent records which have a stronger impact on the patients. Consequently, the learned representations fail to reflect the dynamic trends of patients' prescribing behaviors; (2) Among all dynamic models, the spatio-temporal baselines (i.e., HGT, TGAT, TGN, CTDNE, RxNet) generally obtain better results than LSTM,

Table 1: Overall performance comparison for over-prescribing prediction. Results are reported as mean±std%, the best performance is bolded and runner-ups are underlined. Split ways are denoted as history months/future months (train/val/test).

Metric	Accuracy	Macro-F1	ROC-AUC	Accuracy	Macro-F1	ROC-AUC	Accuracy	Macro-F1	ROC-AUC
Split	9/3 Split (70/10/20)			10/2 Split (70/10/20)			11/1 Split (70/10/20)		
LSTM	65.93±0.81	65.14±1.32	70.65±0.71	64.29±0.43	63.15±1.90	68.29±0.58	62.21±0.88	62.10±0.75	67.02±1.17
GCN	62.31±0.65	61.75±0.68	62.62±0.85	64.05±0.59	63.87±0.54	65.15±0.89	64.50±0.48	64.45±0.67	64.22±0.72
GAT	64.82±0.33	64.67±0.43	65.91±0.90	67.42±0.48	67.18±0.57	70.34±0.95	67.76±0.25	67.42±0.39	70.22±0.86
GraphSage	64.00±0.45	63.75±0.45	65.12±0.95	66.45±0.39	66.21±0.51	69.60±0.77	66.20±0.45	65.50±0.50	67.88±0.84
HGT	65.75±0.68	65.34±0.39	66.04±0.83	67.82±0.86	67.46±0.76	68.73±0.76	68.95±0.44	68.67±0.50	70.26±0.96
TGAT	67.65±0.49	67.42±0.58	70.23±0.94	69.75±0.65	69.73±0.81	70.45±1.10	70.50±0.40	70.35±0.42	71.68±0.80
TGN	69.06±0.38	68.87±0.40	70.56±0.71	70.37±0.61	70.24±0.55	71.57±0.92	71.05±0.59	70.97±0.56	72.42±0.63
CTDNE	66.18±0.56	66.21±0.51	67.78±0.90	68.58±0.74	68.12±0.77	69.48±0.88	67.77±0.48	67.47±0.52	69.21±0.72
JODIE	68.55±0.49	68.42±0.40	69.94±0.67	69.48±0.43	69.25±0.62	70.73±0.64	70.16±0.46	69.68±0.32	70.40±0.90
RxNet	70.73±0.39	70.41±0.42	71.46±0.75	72.62±0.36	72.58±0.33	73.45±0.70	73.38±0.20	73.24±0.26	74.23±0.88
DDHGNN	**74.37±0.45**	**74.06±0.43**	**75.86±0.83**	**76.14±0.52**	**75.90±0.48**	**77.53±0.95**	**77.15±0.39**	**77.13±0.40**	**77.79±0.62**
split	9/3 Split (60/10/30)			10/2 Split (60/10/30)			11/1 Split (60/10/30)		
LSTM	66.10±0.98	66.02±0.54	70.74±0.74	64.46±0.20	63.94±2.22	69.14±0.78	63.35±0.62	63.02±0.90	66.28±1.10
GCN	60.73±0.54	60.32±0.49	60.71±0.68	62.22±0.38	62.05±0.46	62.17±0.60	62.52±0.48	62.08±0.45	62.33±0.70
GAT	61.85±0.42	61.70±0.40	61.88±0.75	65.15±0.41	64.83±0.32	68.77±0.78	64.72±0.56	64.25±0.39	67.50±0.60
GraphSage	60.96±0.38	60.78±0.31	60.66±0.80	63.95±0.58	63.70±0.60	65.46±0.83	64.54±0.65	64.11±0.36	65.18±0.85
HGT	64.15±0.50	64.00±0.56	64.64±0.65	66.42±0.32	66.32±0.35	65.71±0.49	66.28±0.60	66.06±0.45	67.95±0.72
TGAT	66.45±0.51	66.18±0.45	67.06±0.82	68.47±0.45	68.32±0.44	70.14±0.53	68.52±0.60	68.15±0.46	69.42±0.69
TGN	67.92±0.33	67.85±0.28	69.34±0.75	69.33±0.37	69.29±0.34	70.35±0.50	69.94±0.27	69.59±0.33	70.92±0.68
CTDNE	65.82±0.18	65.60±0.26	65.17±0.60	68.21±0.56	67.88±0.52	69.43±0.35	67.73±0.32	67.27±0.52	69.35±0.66
JODIE	67.26±0.44	67.15±0.50	68.20±0.90	68.45±0.46	67.72±0.57	69.90±0.68	68.10±0.52	67.50±0.65	69.26±0.54
RxNet	69.69±0.38	69.52±0.35	70.44±0.52	70.74±0.43	70.16±0.52	71.12±0.66	71.26±0.30	70.48±0.19	72.15±0.57
DDHGNN	**73.58±0.35**	**73.50±0.38**	**74.52±0.77**	**74.95±0.27**	**74.66±0.37**	**75.81±0.45**	**75.52±0.41**	**75.46±0.48**	**76.28±0.41**

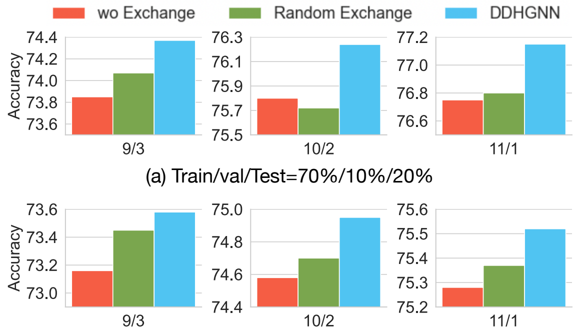
demonstrating the effectiveness of taking spatial information into consideration; (3) DDHGNN achieves stat-of-the-art performance and beats all baselines on all experimental settings, indicating the effectiveness of our proposed model. Additionally, RxNet gains the runner-up performance, the superior performance of DDHGNN and RxNet further prove the importance of preserving the semantics inside a heterogeneous graph.

5.4 Ablation Study

To verify the effectiveness of different modules in DDHGNN, we design five model variants and compare their performance with DDHGNN. The details of these model variants are illustrated below and the comparison results are shown in Table 2.

- **w/o Edge.** The edge features in Equation 2 are discarded.
- **w/o Time.** The temporal information in Equation 2 is discarded.
- **w/o Inter.** The inter-relation aggregation is skipped.
- **w/o Disentangle.** The prior-enhanced adversarial disentangler is removed.

From Table 2, we can find that our model consistently outperforms other variants for all settings. The model variant w/o Edge brings the largest amount of performance deterioration, suggesting the importance of dynamic drug-related information (quantity and days-supply) and patient-related zipcode consistency. We also notice that the performance of the model variant w/o time decreases by a noticeable amount. By comparing DDHGNN and the two variants, the advantage of considering both spatial and temporal information is further revealed. On the other line, the inter-relation aggregation brings considerable performance elevation in that it combines multiple types of relations with learned weights and highlights the more important semantics, which is more sophisticated than average summation. One can clearly see that the disentangle module almost introduces the second largest improvement to DDHGNN, especially for the split settings with a longer historical part. This phenomenon indicates the effectiveness of DDHGNN in collecting distinct and informative factors from entangled representations and the impact is clearer for long-history-dependent predictions.



(a) Train/val/Test=70%/10%/20%

(b) Train/val/Test=60%/10%/30%

Figure 4: performance of different prior exchange ways.

Another set of ablation study is to investigate the effectiveness of our prior-exchange mechanism. The related model variants are explained below and the comparison results are shown in Figure 4.

- **w/o Exchange.** The prior-exchange module is discarded.

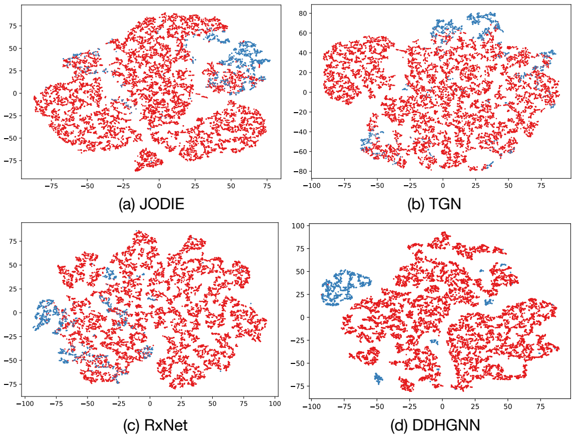
Table 2: Overall comparison of model variants' performance. Results are reported as mean±std%, the best performance is bolded. Split ways are denoted as history months/future months (train/val/test).

Metric	Accuracy	Macro-F1	ROC-AUC	Accuracy	Macro-F1	ROC-AUC	Accuracy	Macro-F1	ROC-AUC
Split	9/3 Split (70/10/20)			10/2 Split (70/10/20)			11/1 Split (70/10/20)		
wo Edge	70.15±0.70	70.02±0.63	69.68±0.87	71.42±0.63	71.28±0.54	71.06±0.71	71.64±0.67	71.29±0.21	71.85±0.87
wo Time	72.23±0.45	72.20±0.32	73.08±0.71	73.85±0.37	73.56±0.41	75.74±0.68	74.24±0.56	74.15±0.55	76.32±0.85
wo Inter	71.88±0.65	71.57±0.59	71.65±0.69	73.06±0.50	72.84±0.45	73.62±0.94	73.87±0.47	73.70±0.38	74.70±0.65
wo Disentangle	72.45±0.59	72.27±0.64	73.64±1.16	73.32±0.56	73.08±0.64	75.20±0.87	74.38±0.19	74.30±0.25	75.49±0.58
DDHGNN	**74.37±0.45**	**74.06±0.43**	**75.86±0.83**	**76.14±0.52**	**75.90±0.48**	**77.53±0.95**	**77.15±0.39**	**77.13±0.40**	**77.79±0.62**
Split	9/3 Split (60/10/30)			10/2 Split (60/10/30)			11/1 Split (60/10/30)		
wo Edge	68.75±0.58	68.50±0.60	68.68±0.48	69.65±0.52	69.58±0.40	70.02±0.87	70.65±0.16	70.60±0.24	71.34±0.25
wo Time	71.21±0.33	71.12±0.45	72.05±0.39	72.88±0.46	72.55±0.67	73.47±0.75	73.38±0.45	73.23±0.50	73.40±0.68
wo Inter	70.24±0.48	70.20±0.12	70.87±0.51	72.54±0.41	72.38±0.26	73.27±0.61	73.17±0.36	73.06±0.42	74.54±0.37
wo Disentangle	70.35±0.38	70.10±0.48	71.54±0.68	72.26±0.58	72.15±0.37	73.75±0.54	72.42±0.65	72.42±0.63	73.62±0.37
DDHGNN	**73.58±0.35**	**73.50±0.38**	**74.52±0.77**	**74.95±0.27**	**74.66±0.37**	**75.81±0.45**	**75.52±0.41**	**75.46±0.48**	**76.28±0.41**

- **Random Exchange.** The priors are exchanged between two random samples, instead of samples within different classes.

The results from Figure 4 show that our model outperforms the two model variants by a noticeable amount, and the variant with random prior exchange achieves better results than the w/o exchange variant in most cases, both of which indicate that our disentangler is able to separate label-irrelevant information from the entangled patient embedding, thus making it more informative.

5.5 Embedding Visualization


(a) JODIE

(b) TGN

(c) RxNet

(d) DDHGNN

Figure 5: TSNE Visualization of positive (blue) and negative (red) patients' embeddings generated by different models.

To further examine our model's capabilities, we visualize the generated patient embeddings using the t-SNE [28] algorithm, as shown in Figure 5. Here, we choose three other dynamic models (i.e., JODIE, TGN and RxNet) for comparison. It can be seen that embeddings of positive and negative patients generated by the two heterogeneous models (i.e., RxNet and DDHGNN) are less overlapped than the rest compared models. Furthermore, the embeddings generated by DDHGNN bring the largest cluster-wise distance for patients with different labels, therefore resulting in better performance.

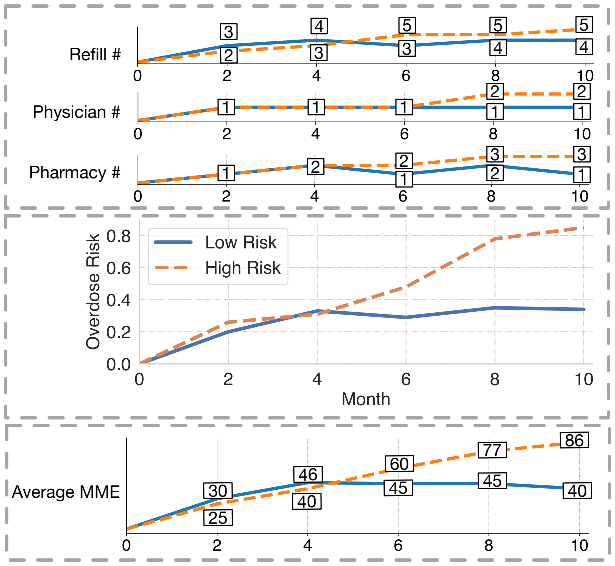


Figure 6: The upper figure is the comparison of refill number, physician number, and pharmacy number of the two patients respectively. The middle figure shows the ten-month overdose risks predicted by DDHGNN for the two patients evaluated every two months. The bottom figure shows the average daily MMEs transformed from their prescriptions every two months.

5.6 Case Study

To evaluate the ability of DDHGNN in capturing prescribing patterns and predicting real-time potential over-prescribing risk, we compare the predicted risk levels of two patients with different labels in our dataset, as shown in Figure 6. Their demographic and medical information is as follows:

- The low-risk patient is a 55-year-old female who fills Hydrocodone & Comb and Tramadol monthly with relatively stable drug doses every time. Her drugs are prescribed by the same physician and she refills her prescriptions at two pharmacies.
- The high-risk patient is a 42-year-old male who was prescribed with Oxycodone & Comb and Morphine Sulfate & Comb. He visits two physicians for the same drug and refills his prescriptions at

three pharmacies. His drug doses are increasing gradually and his daily MME approaches to almost 90 by the end of October.

From Figure 6, we can find that both of the two patients are on a low overdose risk level before April, after which the differences between their risk levels significantly increase. Specifically, the high-risk patient gradually increases his dosage after April and behaves more unusually after June, while the low-risk patient refills her drugs in a relatively stable pattern. The clear difference between the two patients indicates the effectiveness of DDHGNN in detecting suspicious prescribing patterns timely, which is essential for early prediction of overdose patients. Patients predicted by DDHGNN as high-risk can be intervened early with medical advising, thus decreasing their overdose risk.

6 CONCLUSION

In this paper, we study the problem of predicting patients at high risk of opioid overdose, which is of great importance in current public healthcare surroundings. To target this issue, we propose a novel model named DDHGNN, which integrates spatial and temporal dependencies simultaneously while preserving the heterogeneity of the constructed P&D graph via a DHGAT module with a hierarchical aggregation mechanism. Moreover, DDHGNN introduces a prior-enhanced adversarial disentangler to collect factors particular to patients' prescribing patterns for overdose prediction. Extensive experiments are conducted on a 1-year PDMP data to verify the effectiveness of DDHGNN by comparing it with state-of-the-art methods, revealing its promising further in preventing opioid crisis. More experiments about evaluating the performance of DHTGNN on different opioids and hyper-parameter sensitivity are demonstrated in Appendix G and Appendix H.

ACKNOWLEDGMENT

This work is partially supported by the NSF under grants IIS-2209814, IIS-2203262, IIS-2214376, CNS-2203261, CNS-1814825, IIS-2217239, IIS-2027127, and OAC-2218762, the NIJ 2018-75-CX-0032.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.

[2] Xinyu Dong, Sina Rashidian, Yu Wang, Janos Hajagos, Xia Zhao, Richard N Rosenthal, Jun Kong, Mary Saltz, Joel Saltz, and Fusheng Wang. 2019. Machine learning based opioid overdose prediction using electronic health records. In *AMIA*.

[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.

[4] Deborah Dowell, Tamara M Haegerich, and Roger Chou. 2016. CDC guideline for prescribing opioids for chronic pain—United States, 2016. *The Journal of the American Medical Association* (2016).

[5] Ali Mert Ertugrul, Yu-Ru Lin, and Tugba Taskaya-Temizel. 2019. CASTNet: community-attentive spatio-temporal networks for opioid overdose forecasting. In *ECML/PKDD*.

[6] Yujie Fan, Mingxuan Ju, Shifu Hou, Yanfang Ye, Wenqiang Wan, Kui Wang, Yinming Mei, and Qi Xiong. 2021. Heterogeneous Temporal Graph Transformer: An Intelligent System for Evolving Android Malware Detection. In *KDD*.

[7] Yujie Fan, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. 2022. Heterogeneous Temporal Graph Neural Network. In *SDM*.

[8] Yujie Fan, Yiming Zhang, Yanfang Ye, and Xin Li. 2018. Automatic Opioid User Detection from Twitter: Transductive Ensemble Built on Different Meta-graph Based Similarities over Heterogeneous Information Network.. In *IJCAI*.

[9] Sajjad Fouladvand, Jeffery Talbert, Linda P Dwoskin, Heather Bush, Amy Lynn Meadows, Lars E Peterson, Ramakanth Kavuluru, and Jin Chen. 2021. Predicting Opioid Use Disorder from Longitudinal Healthcare Data using Multi-stream Transformer. In *AMIA*.

[10] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*.

[11] Jeffrey Fudin, Mena Raouf, Erica L Wegrzyn, and Michael E Schatman. 2018. Safety concerns with the Centers for Disease Control opioid calculator. *Journal of Pain Research* (2018).

[12] David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling.. In *LREC*.

[13] Rebecca L Haffajee, Anupam B Jena, and Scott G Weiner. 2015. Mandatory use of prescription drug monitoring programs. *The Journal of the American Medical Association* (2015).

[14] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

[15] Dae-Hee Han, Shieun Lee, and Dong-Chul Seo. 2020. Using machine learning to predict opioid misuse among US adolescents. *Preventive Medicine* (2020).

[16] Holly Hedegaard, Arialdi M Miniño, Margaret Warner, et al. 2020. Drug overdose deaths in the United States, 1999-2018. (2020).

[17] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* (1997).

[18] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW*.

[19] Alene Kennedy-Hendricks, Matthew Richey, Emma E McGinty, Elizabeth A Stuart, Colleen L Barry, and Daniel W Webster. 2016. Opioid overdose deaths and Florida's crackdown on pill mills. *American Journal of Public Health* (2016).

[20] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[21] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *KDD*.

[22] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*.

[23] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-time dynamic network embeddings. In *WWW*.

[24] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *AAAI*.

[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.

[26] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. In *ICML*.

[27] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In *WSDM*.

[28] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* (2008).

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[30] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*.

[31] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

[32] Da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. 2020. Inductive representation learning on temporal graphs. In *ICLR*.

[33] Hansheng Xue, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Yu Lin. 2020. Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn. In *ECML/PKDD*.

[34] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

[35] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[36] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD*.

[37] Chuxu Zhang, Ananthram Swami, and Nitesh V Chawla. 2019. Shne: Representation learning for semantic-associated heterogeneous networks. In *WSDM*.

[38] Jianfei Zhang, Ai-Te Kuo, Jianan Zhao, Qianlong Wen, Erin Winstanley, Chuxu Zhang, and Yanfang Ye. 2021. RxNet: Rx-refill Graph Neural Network for Overprescribing Detection. In *CIKM*.

[39] Yiming Zhang, Yujie Fan, Yanfang Ye, Xin Li, and Erin L Winstanley. 2018. Utilizing social media to combat opioid addiction epidemic: automatic detection of opioid users from twitter. In *AAAI*.

[40] Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. 2020. Network Schema Preserved Heterogeneous Information Network Embedding. In *IJCAI*.

[41] Jianan Zhao, Qianlong Wen, Shiyu Sun, Yanfang Ye, and Chuxu Zhang. 2021. Multi-view Self-supervised Heterogeneous Graph Embedding. In *ECML/PKDD*.

A PSEUDO-CODE OF DDHGNN

Algorithm 1: DDHGNN Learning and Prediction

```
Data: Heterogeneous temporal graph $\mathcal{G}_t$, node features $\mathbf{X}$,
      edge features $\mathbf{E}$, layer $L$, priors $\mathbf{o}$, node class labels $\mathbf{Y}$
Result: Probabilities $\widehat{\boldsymbol{y}}$
for each epoch do
    $\mathbf{h}^{0}_{v,t} \leftarrow$ Eq. 1;
    for $l = 1$ to $L$ do
        $\mathbf{h}^{l,r}_{v,t} \leftarrow$ Eqs. 2-3;
        $\mathbf{h}^{l}_{v,t} \leftarrow$ Eqs. 4-6;
    $\mathbf{H} \leftarrow \mathbf{H}^{L}_{T}$
    Feed $\mathbf{H}$ into disentangler by cross-label pair
    for each cross-label pair (i, j) in a batch do
        Sample a batch of real inputs pair
          $(< \mathbf{h}_i, \mathbf{o}_i, y_i >, < \mathbf{h}_j, \mathbf{o}_j, y_j >)$        ▷ $y_i \neq y_j$;
        Generate synthetic inputs for both patients
          $< G_d\left(G_e(\mathbf{h}_i), \mathbf{o}_i, \mathbf{z}\right), \mathbf{o}_i, y_i >$;
        Exchange their priors then generate another pair of
          synthetic inputs $< G_d\left(G_e(\mathbf{h}_j), \mathbf{o}_i, \mathbf{z}\right), \mathbf{o}_i, y_i >$;
        Update $D$ by Eq. 9;
        Freeze $D$;
        Generate a batch of synthetic inputs
          $< G_d\left(G_e(\mathbf{h}_i), \mathbf{o}_i, \mathbf{z}\right), \mathbf{o}_i, y_i >$ and prior-exchange
          synthetic inputs $< G_d\left(G_e(\mathbf{h}_j), \mathbf{o}_i, \mathbf{z}\right), \mathbf{o}_i, y_i >$;
          Update $G$ by Eq. 10
return $G_e(\mathbf{H})$
Compute the prediction loss $L_p$ with $G_e(\mathbf{H})$ via Eqs. 12 and
  update $\Theta$
```

B SELECTED OPIOIDS

The task of our work is to predict patients with high-risk opioid overdose intention, while PDMP contains many records associated with drugs other than opioids. Therefore, records associated with patients who have any historical record related to other medications are eliminated. Specifically, the drug types are limited to Morphine, Hydrocodone, Codeine, Hydromorphone, Oxycodone, Tapentadol, Oxymorphone, Burtorphanol, Pentazocine, Tramadol, Meperidine, Dihydrocodeine, Levorphanol, Fentanyl, Methadone, Buprenorphine and Opium.

C PRIOR COMPUTATION

The priors is generated from three relation views (i.e., patient-physician, patient-pharmacy and patient-drug). Taking patient-physician relation as an example to illustrate how we generate the prior related to this relation. Assume there are N patients in total and the number of physicians that each patient visited at timestamp t can be represented by $\mathbf{M} = [m_1, m_2, \ldots, m_N]$. Then, for patient i, we calculate its patient-physician prior as:

$$o_i^1 = \frac{\sum_{j=1}^{N}\left(m_i \leq m_j \wedge i \neq j\right)}{N} \tag{14}$$

Based on the number of pharmacies and drugs connected to each patient , we can compute the priors specific to patient-pharmacy and patient-drug in a similar way. By concatenating these three priors, we gain the prior vector for patient i, denoted as $\mathbf{o}_i \in \mathbb{R}^{1\times 3}$

D OVERDOSE LABELING

We label our PDMP data based on the definition of Morphine Milligram Equivalents (MME) stressed in [11] and the formulation to compute MME is given as follows:

$$\begin{aligned}\text{MME/day} =& (\text{Number of Units or Days Supply})\times \\ & \text{Strength per Unit} \times \text{MME Conversion Factor}\end{aligned} \tag{15}$$

Since Centers for Disease Control and Prevention (CDC) recommends patients to avoid or carefully justify increasing dosage to 90 MME/day [11], we make the following definitions: (1) Once a prescription is filled, the converted daily MME will be added to the patient's daily MME amount for the next *days-supply* days; (2) Within the duration, if the patient is prescribed with another drug, the daily MME is added to the overlapping days; (3) A patient is considered at high risk of over-prescribing if his/her average daily MME exceeds 90 for 4 straight days, or the patient has over 8 days with a daily MME over 90 in a month. We calculate the daily MMEs for each patient on each day of 2016 and label their over-prescribing risks following the above definitions. Given the label definition rule, we conduct experiments on three *history-future* split variants and label the data based on the records of last 3, 2 and 1 month(s). The statistics of the three split variants are provided in Table 3

Table 3: History-Future Split Patient Label Distribution

Split Ratio	9/3	10/2	11/1
Negative	258506	258048	257773
Down-sampled Negative	10711	6711	3507
Positive	10711	6711	3507

E BASELINE IMPLEMENTATION

Among the baselines of this work, HGT and RxNet are originally used for dynamical heterogeneous graph, thus we can almost directly apply them to PDMP data. However, other baselines are not proposed for dynamical heterogeneous graph, therefore we make some specific adaptions for these baselines in our experiments. Particularly, we transform all historical prescription records of each patient to sequential inputs for LSTM, and each prescription represents a token in the sequence. Due to the reason that patients have different number of records, we pad the sequence inputs of all the patients to the same length. For the static GNN models (i.e., GCN, GAT, GraphSage), we first transform the heterogeneous graph into homogeneous graph and then remove the temporal information in the dynamic graph to run the experiment. For the dynamic graph learning baselines designed for homogeneous graph (i.e., CTDNE, TGAT, TGN, JODIE), we simply transform the heterogeneous graph into homogeneous and follow the node classification case provided by their original implementation.

F PARAMETER SETTING

During the experiments, we fix some hyper-parameters for the convenience of tuning work: the depth of all GNN models is set to 2, the weight decay is set to 1e-5. Other import hyper-parameters are tuned on each split by grid search. Specifically, we search learning rate in $\{0.001, 0.005, 0.01, 0.05\}$, batch size in $\{32, 64, 128\}$, embedding hidden dimension d in $\{16, 32, 64, 128, 256\}$, head number K in $\{1, 2, 4, 8, 16\}$, dropout in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$, balance coefficient λ in $\{0.1, 0.5, 1.0\}$, to obtain best results of DDHGNN and all the baselines.

G PERFORMANCE OVER DIFFERENT DRUGS

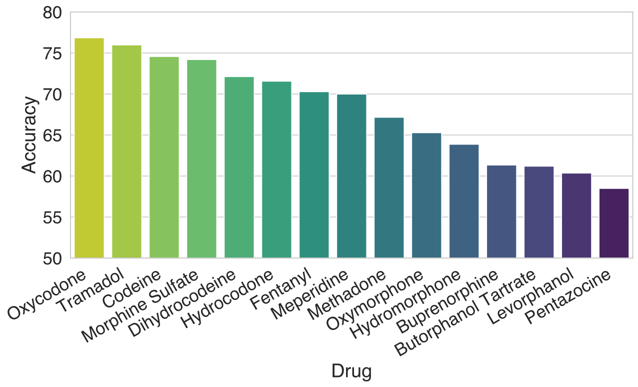


Figure 7: Accuracy over different drugs in percentage.

To further estimate our model over different drugs, we show the predicted results in terms of accuracy for each drug group in Figure 7, where all the patients in each drug group have at least one corresponding prescribing record. We run our model for the prediction task over each group of patients for evaluation. From Figure 7, one can clearly see that there are variations among the performance of different drug groups, where the commonly used opioids in PDMP, like Oxycodone, Tramadol and Codeine, achieve better performance than other less prescribed opioids. The better performance over the popular opioids indicates the ability of our model to detect drug crisis (i.e., the related-death and hospitalization number increase with the popularity of a certain opioid).

H PARAMETER SENSITIVITY ANALYSIS

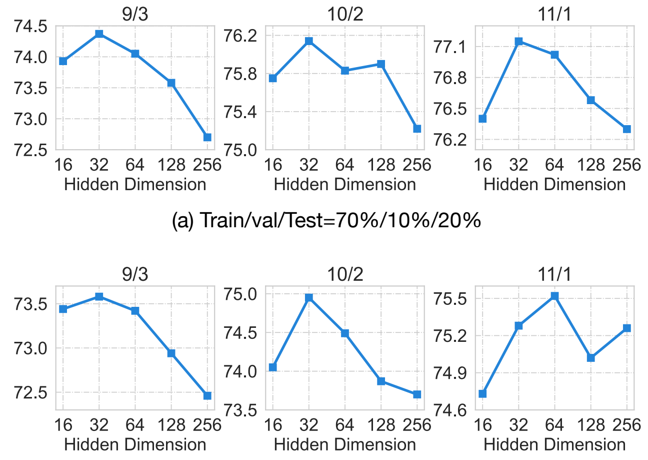
In this section, we study the performance sensitivity of the embedding hidden dimension d and the number of attention head K on the proposed DDHGNN.

H.1 Embedding Dimension

We first investigate the effect of the dimension of the hidden embedding d on performance, the search space is stated in Appendix F and we demonstrate the experimental results in Figure 8. We can find that d=32 is the optimal embedding hidden dimension value for most of the settings. A common phenomena found is that the performance of our model first increases and then decreases as the embedding dimension increases. The reason is that DDHGNN needs a suitable dimension to encode the informative factors and a larger dimension may introduce additional redundancies.

H.2 Number of Attention Head

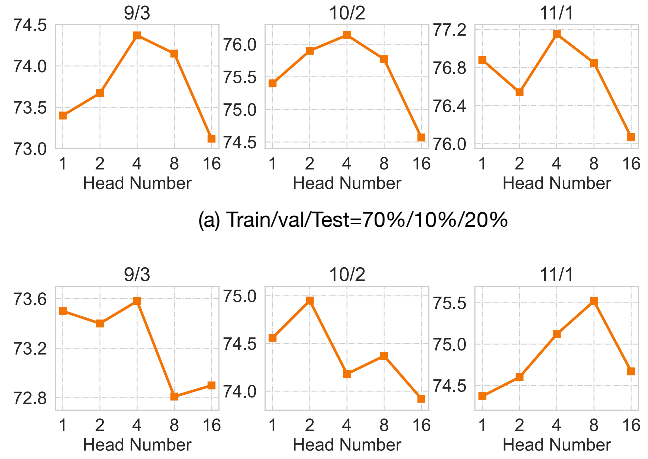
We further check the influence of the number of attention head K on the prediction performance. We vary K in the search space introduced in 8 and exhibit the results in Figure 9. One can see that none of the settings achieves the best performance when K=1, indicating multi-head attention mechanism generally improve the performance of DDHGNN. Additionally, the best values of K for different settings are not exactly same, therefore the value of K should be carefully tuned to achieve optimal performance.



(a) Train/val/Test=70%/10%/20%

(b) Train/val/Test=60%/10%/30%

Figure 8: Prediction performance in accuracy of DDHGNN w.r.t. different hidden embedding dimension.



(a) Train/val/Test=70%/10%/20%

(b) Train/val/Test=60%/10%/30%

Figure 9: Prediction performance in accuracy of DDHGNN w.r.t. different number of attention head.