Customized Subgraph Selection and Encoding for Drug-drug Interaction Prediction

Haotong Du[1] **Quanming Yao**[2] **Juzheng Zhang**[2] **Yang Liu**[1] **Zhen Wang**[1†]
[1]Northwestern Polytechnical University [2]Tsinghua University
duhaotong@mail.nwpu.edu.cn w-zhen@nwpu.edu.cn
qyaoaa@tsinghua.edu.cn {juzhengzh00,yangliuyh}@gmail.com

Abstract

Subgraph-based methods have proven to be effective and interpretable in predicting drug-drug interactions (DDIs), which are essential for medical practice and drug development. Subgraph selection and encoding are critical stages in these methods, yet customizing these components remains underexplored due to the high cost of manual adjustments. In this study, inspired by the success of neural architecture search (NAS), we propose a method to search for data-specific components within subgraph-based frameworks. Specifically, we introduce extensive subgraph selection and encoding spaces that account for the diverse contexts of drug interactions in DDI prediction. To address the challenge of large search spaces and high sampling costs, we design a relaxation mechanism that uses an approximation strategy to efficiently explore optimal subgraph configurations. This approach allows for robust exploration of the search space. Extensive experiments demonstrate the effectiveness and superiority of the proposed method, with the discovered subgraphs and encoding functions highlighting the model's adaptability.

1 Introduction

Precise prediction of drug-drug interactions (DDIs) is essential in biomedicine and healthcare research [1]. Drug combination therapy [2] can enhance treatment effectiveness for certain diseases; however, it also increases the risk of adverse drug reactions, potentially threatening patient safety [3]. Identifying DDIs through laboratory experiments is both costly and time-consuming [4, 5]. With the success of deep learning, researchers have increasingly explored computational methods for DDI prediction. Early approaches primarily relied on molecular fingerprint information [6] or hand-engineered features [7], often neglecting the pre-existing interaction properties between drugs.

Considering drugs as nodes and their interactions as edges, DDI prediction can be framed as a multi-relational link prediction problem within the constructed drug interaction network. Recent advancements in graph neural networks (GNNs) [8, 9, 10, 11] have consistently achieved superior performance in this task. Specifically, subgraph-based methods, such as SumGNN [12], EmerGNN [13], and KnowDDI [14], have shown promising results by selecting subgraphs around query edges and applying sophisticated encoding functions (message passing functions) to represent these subgraphs, Such methods transform the multi-relational link prediction task into a multi-type subgraph classification problem. Figure 1 illustrates the pipeline of subgraph-based methods.

However, due to the dense nature [15, 16] of drug interaction networks and their complex interaction semantics [17], existing hand-designed subgraph methods often fail to capture the nuanced but crucial information across different data inputs. In the initial phase of the reasoning pipeline, the subgraph sampler must have the capability to customize the selection of drug subgraphs for different queries, thereby ensuring precise contextualization of the reasoning evidence.

†Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3478

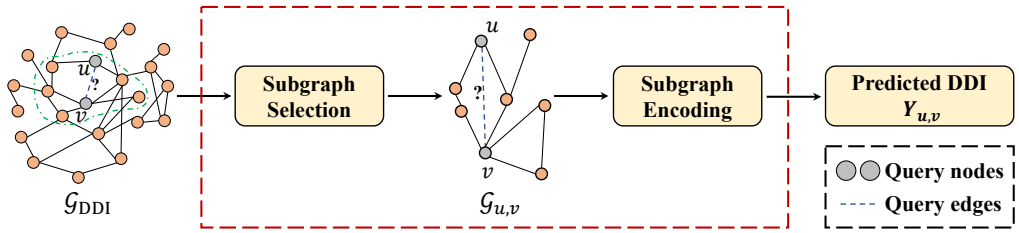


Figure 1: The pipeline of subgraph-based methods includes subgraph selection and subgraph encoding. In this work, we focus specifically on searching for components within the red-dotted lines.

Without customized subgraph selection, SumGNN samples subgraphs using a fixed subgraph range k, selecting the k-hop neighbors of each drug as associated subgraphs for predicting drug-drug interactions (DDIs). This coarse-grained approach is straightforward and easy to implement, but it may introduce noise or omit valuable information needed to reason about diverse drug pair interactions.

Table 1: Comparing with existing methods."-" represents not applicable.

Method	Fine-grained Subgraph Selection	Data-specific Encoding Function
SEAL [18]	✗	✗
GraIL [19]	✗	✗
SumGNN [12]	✗	✗
SNRI [20]	✗	✗
KnowDDI [14]	✓	-
MR-GNAS [21]	-	✓
AutoGEL [22]	-	✓
CSSE-DDI	✓	✓

In terms of encoding process, the encoding function must be capable of modeling a wide variety of drug interactions within the drug interaction network. Real-world drug interactions exhibit complex mechanisms, for instance, metabolism-based interactions display asymmetric semantic patterns, whereas phenotype-based interactions are symmetric. Manually designed encoding functions are limited in their ability to accommodate both types of distinct semantic patterns simultaneously [23]. Therefore, designing a customized and data-adaptive subgraph-based pipeline is essential for effective DDI prediction.

Neural architecture search (NAS) [24, 25] has achieved remarkable success in designing data-specific models, often surpassing architectures crafted by human experts in various fields, such as computer vision [26], graph neural network [27], and knowledge graph learning [23]. However, effectively selecting suitable subgraphs from the vast space of candidates and efficiently optimizing the joint search process of subgraph selection and encoding remain open challenges.

In this paper, we leverage NAS to search for data-specific components in the subgraph-based pipeline. Specifically, we design search spaces for pipeline components, including subgraph selection and encoding spaces, to capture various drug interaction patterns. To enable efficient exploration of the extensive subgraph selection space, we introduce a relaxation mechanism that continuously selects subgraphs in a structured manner. Additionally, we propose a subgraph representation approximation strategy to reduce the high cost of explicit subgraph sampling, enabling efficient and robust search. Compared with existing methods in Table 1, our proposed **C**ustomized **S**ubgraph **S**election and **E**ncoding for **D**rug-**D**rug **I**nteraction prediction (**CSSE-DDI**) achieves fine-grained subgrpah selection and data-specific encoding functions, providing an efficient and precise method for drug interaction prediction. Our main contributions are summarized as follows:

- We present CSSE-DDI, a searchable framework for DDI prediction that adaptively customizes the subgraph selection and encoding processes. To the best of our knowledge, this is the first application of NAS techniques to tailor an adaptive subgraph-based pipeline for the DDI prediction task.
- We construct expressive search spaces to ensure precise capture of evidence for drug interaction prediction. Additionally, we devise a relaxation mechanism to transform the discrete subgraph selection space into a continuous form, enabling differentiable search. Simultaneously, we apply a subgraph representation approximation strategy to mitigate the inefficiencies of explicit subgraph sampling, thereby accelerating the search process.
- Extensive experiments on benchmark datasets demonstrate that our method, which searches for customized pipelines, achieves superior performance compared to hand-designed methods. Additionally, our approach effectively captures the underlying biological mechanisms of drug-drug interactions.

2 Related Works

Subgraph-based Link Prediction Recently, subgraph-based methods [18, 19, 28] have emerged as a promising approach, showing superior performance in link prediction tasks. Different from canonical GNNs, subgraph-based method extracts a subgraph patch for each training and test query, learning a representation of the extracted patch for final prediction, as illustrated in Figure 1.

Existing works has primarily focused on designing more informative subgraphs and more expressive encoding functions. However, they do not take into account customizing these components to deal with various data. Specifically, in terms of subgraph sampler, current approaches lack fine-grained and adaptive extraction for different query subgraphs. While PS2 [29] demonstrates the effectiveness of identifying optimal subgraphs for each edge in homogeneous graph link prediction, there is no comparable work in multi-relational graph link prediction. In dense DDI networks, fine-grained identification of subgraph for different queries is even more crucial.

As for the encoding function, existing works overlook the importance of data-specific encoding, which has been emphasized in recent literature [30, 27]. Customized encoding functions are especially advantageous for drug interaction networks with complex and diverse interactions.

GNN-based DDI Prediction Recently, there has been growing interest in applying GNNs for DDI prediction [8, 9]. However, these works execute message-passing functions over the entire graph, which limits their ability to capture explicit local evidence for specific query drug pairs and lack interpretability. In contrast, subgraph-based DDI prediction methods [12, 63, 13, 14] transform the multi-relational link prediction problem into a subgraph classification problem by extracting subgraphs around query nodes, achieving strong performance. Nevertheless, these works use the same subgraph extraction strategy for all queries and rely on a fixed message-passing function to handle complex DDIs, which limits their flexibility and adaptivity in dense DDI networks.

Graph Neural Architecture Search Graph neural architecture search (GNAS) [31] aims to find high-performing GNN architectures using NAS techniques. Recent studies [30, 27] have explored GNAS to create more expressive GNN models across various tasks. AutoDDI [32], for instance, automatically designs GNN architectures to learn molecular graph representations of drugs for DDI prediction. However, research on optimizing graph sampling for GNAS remains limited due to the diversity of graph-structured data.

Regarding search strategy, early approaches explores the search space using reinforcement learning [33] or evolutionary algorithms [34], which is highly inefficient. One-shot approaches [35] instead construct an over-parameterized network (supernet) and optimize it using gradient descent, leveraging continuous relaxation of the search space to improve search efficiency. The recently proposed few-shot NAS paradigm [36] further enhances supernet evaluation consistency by generating multiple sub-supernets.

3 Proposed Method

3.1 Problem Formulation

Given a set of drugs $\mathcal{V}$ and interaction relations $\mathcal{R}$ among them, the drug interaction network is denoted as $\mathcal{G}_{\text{DDI}} = \{(u, r, v) \mid u, v \in \mathcal{V}, r \in \mathcal{R}\}$, with each tuple (u, r, v) describes an interaction between drug u and drug v. Consequently, drug-drug interaction (DDI) prediction can be framed a multi-relational link prediction task within the drug interaction network $\mathcal{G}_{\text{DDI}}$. The objective is to predict the types of interactions between two given drug nodes, which can be denoted as a query $(u, ?, v)$, i.e., given the query drug-pair entities u and v, to determine the interaction r that makes (u, r, v) valid.

Moreover, instead of directly predicting on the entire graph $\mathcal{G}_{\text{DDI}}$, subgraph-based methods decouple the prediction process into two stages: (1) selecting a query-specific subgraph and (2) encoding the subgraph to predict interactions, as shown in Figure 1. The prediction pipeline then becomes

$$\mathcal{G}_{\text{DDI}} \xmapsto{\texttt{Selection},(u,v)} \mathcal{G}_{u,v} \xmapsto{\texttt{Encoding}} \boldsymbol{Y}_{u,v}, \tag{1}$$

where the sampler selects a subgraph $G_{u,v}$ conditioned on the given query $(u, ?, v)$. Using this subgraph $G_{u,v}$, the encoding function produces the final predictions $\boldsymbol{Y}_{u,v}$.

Building on previous analysis and existing research, and inspired by NAS, we propose to search for data-adaptive subgraph selection and encoding components to obtain a customized subgraph

 https://doi.org/10.52202/079017-3478

pipeline. In Section 3.2, we first introduce the well-designed subgraph selection and encoding spaces to ensure comprehensive coverage of cricual information in various drug interaction networks. Further, in Section 3.3 we present a subgraph relaxation strategy and approximation mechanisms for subgraph representations to facilitate efficient differentiable search. Finally, we develop a robust search algorithm to address the customized search problem with stability and precision.

3.2 Search Space

3.2.1 Subgraph Selection Space

In practice, subgraph-based methods define the drug-pair subgraph between drug pairs as the union or interaction of k-hop ego-network [2] of query drugs. Here, k is a key hyperparameter that determines the range of message propagation aggregated by the central node. Selecting k is crucial to model performance, as it dictates whether the model has access to high-quality evidence context for accurate prediction.

Prior works [12, 37] typically utilize a fixed hyperparameter for all drug pairs, i.e., selecting the union of a fixed k-hop ego-network for arbitrary queries. Nevertheless, this approach can lead to an imprecise collection of evidence for interaction reasoning, potentially undermining the reasoning process due to missing critical information or the inclusion of excessive irrelevant information.

Based on the above analysis, we define a drug-pair subgraph selction space containing a range of subgraphs of different sizes for a given query (u, v):

$$\mathcal{S}_{u,v} = \{\mathcal{G}_{u,v}^{i,j} \,|1 \leq i, j \leq \eta\}, \tag{2}$$

where $\mathcal{G}_{u,v}^{i,j}$ is generated by taking the union of the i-hop ego-network of node u and the j-hop ego-network of node v, i.e., $\mathcal{G}_{u,v}^{i,j} = \{z \in \mathcal{V} \mid z \in (u \cup \mathcal{N}_i(u) \cup v \cup \mathcal{N}_j(v))\}$, where $\mathcal{N}_i(u)$ and $\mathcal{N}_j(v)$ are the i-hop and the j-hop neighbors of u and v, respectively. The threshold η constrains the maximum subgraph range.

Since each drug-pair has a specific subgraph selection space, the overall size of space in the entire graph is $\eta^{2|\mathcal{E}|}$, where $|\mathcal{E}|$ represents the number of edges in the drug interaction network. A larger $|\mathcal{E}|$ result in a subgraph selection space that grows exponentially with the number of edges. Therefore, efficiently searching for the optimal subgraph configurations for different queries is challenging.

3.2.2 Subgraph Encoding Space

For the automated design of the subgraph encoding function, we first adopt a unified message passing framework [21, 38] comprising several key modules: the message-computing function `MES`, the aggregation function `AGG`, the combination function `COM`, and the activation function `ACT`, as follows:

$$\begin{aligned} &\texttt{step 1:}\ \mathbf{m}_u \leftarrow \texttt{AGG}(\texttt{MES}(\mathbf{h}_v, \mathbf{h}_{r(u,v)})_{v \in \mathcal{N}_1(u)}), \\ &\texttt{step 2:}\ \mathbf{h}_u \leftarrow \texttt{ACT}(\texttt{COM}(\mathbf{h}_u, \mathbf{m}_u)), \end{aligned} \tag{3}$$

where $\mathbf{h}_u \in \mathbb{R}^d$ and $\mathbf{h}_r \in \mathbb{R}^d$ represent the embeddings of node u and interaction r, respectively, and $\mathbf{m}_u$ is the intermediate message representation of u aggregated from its neighborhood $\mathcal{N}_1(u)$.

A substantial amount of literature [39, 40, 41] has focused on manually designing these modules to improve performance. However, such encoding functions are inflexible for handling diverse interaction patterns across different drug interaction network. For example, interactions in DrugBank [42] describe how one drug affects the metabolism of another one. The excretion of Acamprosate, for instance, may be decreased when combined with Acetylsalicylic acid (Aspirin). Such interaction pattern is asymmetric, meaning $r(x, y) \nRightarrow r(y, x)$. Conversely, interactions in the TWOSIDES dataset [43] are primarily at the phenotypic level, such as headache or pain in throat, representing symmetric patterns where $r(x, y) \Rightarrow r(y, x)$. These two relational semantics are distinctly different, and existing hand-designed encoding functions struggle to capture such diverse semantics effectively [44, 23].

Here, we aim to perform an adaptive searching for the encoding function in the context of drug interaction prediction. Based on the framework presented in Eq. (3), we design an expressive subgraph encoding space with a set of candidate operations. Detailed explanations of these modules can be found in the Appendix A.1.

[2] A k-hop ego-network of a node consists of the node and its k-hop neighbors.

After encoding the subgraph $\mathcal{G}_{u,v}$, we obtain the representation $\mathbf{z}_{u,v}$ of the input subgraph $\mathcal{G}_{u,v}$. The predictor then maps the representation $\mathbf{z}_{u,v}$ to the probability logits for different interactions between drug pairs, i.e., $y_{u,v} = \mathbf{W}_{\text{pred}}\mathbf{z}_{u,v}$, where $\mathbf{W}_{\text{pred}} \in \mathbb{R}^{2d\times|\mathcal{R}|}$ is the parameter of the predictor.

3.3 Search Strategy

3.3.1 Search Problem

Based on the well-designed search space described above, we formulate a bi-level optimization problem to adaptively search for the optimal configuration of subgraph-based pipelines.

Definition 1 (Customized Subgraph-based Pipeline Search Problem)**.** *Let $\mathcal{A}$ denote the subgraph encoding space, $\mathcal{S}_{u,v}$ represent the subgraph selection space for the query (u, v), $\boldsymbol{\alpha}$ be a candidate encoding function in $\mathcal{A}$, $\mathbf{W}$ represent the parameters of a model from the search space, and $\mathbf{W}^*(\mathcal{G}_{u,v}; \boldsymbol{\alpha})$ denote the trained operation parameters. Let $\mathcal{D}_{\text{tra}}$ and $\mathcal{D}_{\text{val}}$ denote the training and validation sets, respectively. The search problem is formulated as follows:*

$$\arg\max_{\boldsymbol{\alpha}\in\mathcal{A},\mathcal{G}_{u,v}\in\mathcal{S}_{u,v}} \sum\nolimits_{(u,r,v)\in\mathcal{D}_{\text{val}}} \mathcal{M}(\mathbf{W}^*(\mathcal{G}_{u,v};\boldsymbol{\alpha}); \mathcal{G}_{u,v}; \boldsymbol{\alpha}), \tag{4}$$

$$s.t.\ \mathbf{W}^*(\mathcal{G}_{u,v};\boldsymbol{\alpha}) = \arg\min_{\mathbf{W}} \sum\nolimits_{(u,r,v)\in\mathcal{D}_{\text{tra}}} \mathcal{L}(\mathbf{W}; \mathcal{G}_{u,v}; \boldsymbol{\alpha}), \tag{5}$$

where the classification loss $\mathcal{L}$ is minimized for all interactions, while the performance measurement $\mathcal{M}$ is expected to be maximized.

In this work, we adopt the differentiable search paradigm [45] to solve the bi-level optimization problem, which is widely used in recent NAS literature [46] and enables efficient exploration of the search space. Nevertheless, our proposed subgraph selection space poses two technical challenges: **First**, we cannot directly apply relaxation strategies, which is a prerequisite for differentiable NAS methods, to make the discrete selection space continuous. This limitation arises because different subgraphs in the selection space contain diverse nodes and edges, making it challenging to design a relaxation function that unifies subgraphs of varying sizes. **Second**, to enable searching within the subgraph selection space, we would need to first generate all subgraphs in the space. However, sampling such a large number of subgraphs is computationally intractable.

To address these challenges, we design a subgraph selection space relaxation mechanism in Section 3.3.2 . Additionally, we introduce an intuitive subgraph representation approximation strategy in Section 3.3.3 to reduce the high costs associated with explicit sampling.

3.3.2 Relaxation of Subgraph Selection Space

Technically, as in existing NAS works [45, 47], one typically needs to relax the search space into continuous form to enable effective backpropagation training. However, for the subgraph selection space, the traditional continuous relaxation strategy is not directly applicable due to the structural mismatch between graphs and vectors.

To address this, we first utilize encoding function $f(\cdot)$ to encode subgraphs with different scopes. This approach provides all subgraphs with representations of the same dimension, making it feasible to implement a relaxation strategy. Additionally, inspired by the reparameterization trick [48], we adopt the Gumbel-Softmax function to facilitate differentiable learning over a discrete space:

$$\hat{\mathbf{z}}^{i,j}_{u,v} = \sum_{1\le i,j\le\eta} \frac{\exp(\log(g(f(\mathcal{G}^{i,j}_{u,v})) + \mathbf{G}_{i,j})/\tau)}{\sum_{i',j'=1}^{\eta}\exp(\log(g(f(\mathcal{G}^{i',j'}_{u,v})) + \mathbf{G}_{i',j'})/\tau)} f(\mathcal{G}^{i,j}_{u,v}), \tag{6}$$

where $g(\cdot)$ scores the subgraph representations using multiple linear layers, $\mathbf{G}_{i,j} = -\log(-\log(\mathbf{U}_{i,j}))$ is the Gumbel random variable, $\mathbf{U}_{i,j}$ is a uniform random variable, and τ is the temperature parameter controlling sharpness. $\hat{\mathbf{z}}^{i,j}_{u,v}$ is the mixed selection operation of subgraph $\mathcal{G}^{i,j}_{u,v}$ used to optimize searching process.

3.3.3 Subgraph Representation Approximation Strategy

To solving the optimization problem as Eq. (4) and (5), we need to explicitly sample all the candidate subgraphs within the subgraph selection space $\mathcal{S}_{u,v}$ for each query. However, one of the most challenging aspects of subgraph-based approaches is their inefficient subgraph sampling process [49, 50, 51].

https://doi.org/10.52202/079017-3478

Upon examining our subgraph selection space, we observe that all subgraphs are generated by combining multi-hop ego-networks of the target nodes, encompassing multiple neighborhood hops. Inspired by the k-subtree extractor [52], we apply an encoding function to the entire graph and use the resulting node representations of u and v as the ego-network representations for these nodes. The representation of the drug pair can then be obtained by concatenating the ego-network representations of u and v. Formally, if we denote by $f(\mathcal{G}_{\text{DDI}}, u, i)$ the i-layer hidden representation of node u produced by encoding function applied to $\mathcal{G}_{\text{DDI}}$, then

$$f(\mathcal{G}_{u,v}^{i,j}) \approx \texttt{CONCAT}(f(\mathcal{G}_{\text{DDI}}, u, i), f(\mathcal{G}_{\text{DDI}}, v, j)), \tag{7}$$

The k-subtree extractor represents the k-subtree structure rooted at a given node, which mirrors the structure as the k-hop ego-network. This approximation strategy only requires executing the encoding function on the entire drug interaction network, thereby efficiently yielding subgraph representations of varying scopes, which significantly improves the efficiency in solving the bi-level optimization problem.

3.3.4 Robust Search Algorithm

Using the proposed subgraph selection relaxation mechanism, we can transform the overall discrete search space in Definition. 1 into a continuous form, allowing the search problem to be solved by the one-shot NAS paradigm. Additionally, our subgraph representation approximation strategy efficiently obtains subgraph representations and reduces search costs

Following [53], we adopt the single path one-shot training strategy (SPOS) [54] to reduce the computational cost of supernet training. However, the one-shot approach [55, 56, 45], i.e., using the same supernet parameters $\mathbf{W}$ for all architectures, can decrease the consistency between the supernet's performance estimation and the ground-truth performance [57]. Inspired by few-shot NAS [36], we propose a message-aware partitioned supernet training strategy to mitigate the coupling effect of different message-computing operators [58]. By partitioning the superent to form sub-supernets based on the type of message-computing function, this strategy improves the consistency and accuracy of supernet, enabling the search algorithm more stable and robust. Algorithm 1 delineates the full procedure, with further details provided in Appendix A.2.

Algorithm 1: The search algorithm of CSSE-DDI.

Input: Supernet $\mathcal{S}$, number of partitions based on message computing function categories M $(M = 4)$, subsupernet $\mathcal{S}_i, (i = 1, \cdots, M)$.

`// supernet training phase`

Train $\mathcal{S}$ by continuously sampling a single path until convergence;

`// supernet partition phase`

Partition $\mathcal{S}$ into M sub-supernets $\mathcal{S}_1, \cdots, \mathcal{S}_M$;

`// sub-supernet training phase`

forall $i = 1, \cdots, M$ **do**

  Initialize $\mathcal{S}_i$ with weights transferred from $\mathcal{S}$;

  Train $\mathcal{S}_i$ by continuously sampling a single path until convergence;

end

`// searching phase`

Search the optimal encoding function from sub-supernets $\mathcal{S}_1, \cdots, \mathcal{S}_M$ on validation data by natural gradient descent;

Select the optimal subgraphs from sub-supernets $\mathcal{S}_1, \cdots, \mathcal{S}_M$ on validation data by preserving the subgraphs with the largest probabilities;

3.4 Comparison with Existing Works

While many works [12, 13, 14] have explored DDI prediction using subgraph-based methods, our approach introduces two significant advancements. First, to the best of our knowledge, our method (CSSE-DDI) is the first to customize the subgraph selection and encoding processes specifically for subgraph-based DDI prediction. In contrast, previous methods rely on fixed subgraph selection strategy to sample subgraphs and employ hand-designed functions for encoding, as summarized in Table 1. Consequently, our method can adapt data-specific components within subgraph-based pipelines, outperforming existing methods in both performance and efficiency (Section 4.2). Moreover, our approach not only selects fine-grained drug-pair subgraphs that enhance interpretability through potential pharmacokinetic and metabolic concepts (Section 4.6.1), but also searches for data-specific encoding functions that accurately capture the semantic features of drug interactions (Section 4.6.2).

4 Experiments

4.1 Experimental Setup

Datasets Experiments are conducted on two public benchmark DDI datasets: DrugBank [42] and TWOSIDES [43]. Detailed descriptions of these datasets are presented in Appendix B.1.

Experimental Settings Following [13], we examine two DDI prediction task settings: S0 and S1. Let the drug pairs for DDI prediction be denoted as (u, v). In the S0 setting, both drug nodes u and v are present in the known DDI graph. Existing DDI prediction methods are typically evaluated in this setting. In contrast, the S1 setting involves a pair (u, v) where one drug is known and the other is a novel drug not represented in the known DDI graph. This scenario highlights the critical need for DDI predictions involving new drugs in real-world applications.

Evaluation Metric We follow [12] to evaluate our method. For the DrugBank dataset, where each drug pair contains only one interaction, we use the following metrics: F1 Score, Accuracy and Cohen's κ. For the TWOSIDES dataset, where multiple interactions may exist between a pair of drugs, we consider the following metrics: ROC-AUC, PR-AUC and AP@50. Additional details are provided in Appendix B.2.

Baselines We compare CSSE-DDI with the following representative DDI prediction method: (i) GNN-based methods include Decagon [8], GAT [59], SkipGNN [9], CompGCN [60], ACDGNN [61], and TransFOL [62]. (ii) Subgraph-based methods include SEAL [18], GraIL [19], SumGNN [12], SNRI [20], KnowDDI [37] and LaGAT [63]. (iii) NAS-based method include MR-GNAS [21], and AutoGEL [22].

We also compare our method with two variants, including CSSE-DDI-FS and CSSE-DDI-FF. The configurations of these variants are as follows: (i) **CSSE-DDI-FS**: This variant omits fine-grained subgraph selection for each query, using fixed k-layer drug node representations to generate the subgraph representation. (ii) **CSSE-DDI-FF**: This variant does not search for the encoding function, instead using a fixed encoding function backbone to capture semantic and topological features in the drug interaction network. In this case, we employ a 3-layer CompGCN model as the backbone. For all baselines, we obtain the results by rerunning the released codes.

Implementation We implement our method[3] based on PyTorch framework [64]. Following existing GNN-based methods [37], we select a 3-layer encoding function backbone for both datasets. The maximum threshold η for the subgraph selection space is set to 3. More experimental details are given in the Appendix B.3.

4.2 Performance Comparison in S0 settings

Table 2 shows the overall results across all benchmarks in S0 setting. As can be seen, CSSE-DDI consistently outperforms all baselines on each dataset, demonstrating its effectiveness in searching for data-specific subgraph-based pipelines for DDI prediction task. Among the baselines, subgraph-based methods significantly outperform full-graph-based methods due to their enhanced ability to reason over local subgraph contexts. Within the subgraph-based methods, SEAL, GraIL, SumGNN, and SNRI use a fixed sample strategy to select subgraphs, which may not be optimal for different drug-pair queries.

When it comes to NAS-based method, MR-GNAS and AutoGEL contain well-established search spaces that embrace multi-relational message-passing schema, focusing primarily on automated encoding function design using the one-shot NAS paradigm. While CSSE-DDI adopts a single path supernet training strategy and a message-aware partitioning approach to search for data-adaptive subgraph-based pipelines with stability and robustness, enabling the model to achieve excellent performance across various datasets. Moreover, the consistent performance gains of CSSE-DDI over its two variants validate the importance of jointly customizing subgraph-based pipeline components, i.e., fine-grained subgraphs and data-specific encoding functions, to fit datasets rather than relying on a fixed approach.

Figure 2 shows the learning curves of several competitive methods on both datasets, including CompGCN, KnowDDI and the proposed CSSE-DDI. As can be seen, the searched models not only outperform the baselines but also demonstrate a clear advantage in efficiency, highlighting that enhancing model flexibility and adaptability is essential for improving performance and efficiency.

[3] Our code is available at https://github.com/LARS-research/CSSE-DDI.

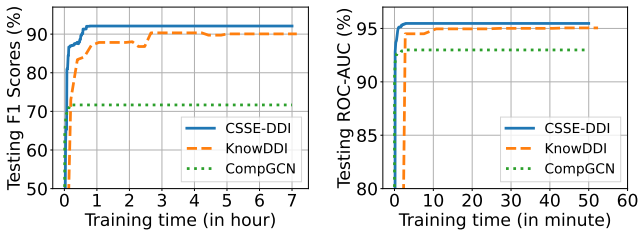
https://doi.org/10.52202/079017-3478

Table 2: CSSE-DDI achieves the best predictive performance compared to state-of-the-art baselines in DDI prediction. Average and standard deviation of five runs are reported. For these metrics, higher values always indicate better performance.

Model Type	Dataset	Dataset 1: DrugBank			Dataset 2: TWOSIDES		
	Task Type	Multi-class			Multi-label		
	Methods	F1 Score	Accuracy	Cohen's κ	ROC-AUC	PR-AUC	AP@50
GNN-based	Decagon	57.35±0.26	87.19±0.28	86.07±0.08	91.72±0.04	90.60±0.12	82.06±0.45
	GAT	33.49±0.36	77.18±0.15	74.20±0.23	91.18±0.14	89.86±0.05	82.80±0.17
	SkipGNN	59.66±0.26	85.83±0.18	84.20±0.16	92.04±0.08	90.90±0.10	84.25±0.25
	CompGCN	71.20±0.70	88.30±0.29	86.15±0.35	93.00±0.07	91.26±0.07	86.18±0.10
	ACDGNN	86.24±0.93	90.53±0.38	87.81±0.33	93.69±0.47	92.12±0.21	87.45±0.24
	TransFOL	89.97±1.64	91.92±0.89	90.92±0.72	94.16±0.62	93.52±0.53	88.13±0.39
Subgraph-based	SEAL	48.82±0.98	76.61±0.26	71.91±0.59	90.74±0.22	90.11±0.17	84.13±0.13
	GraIL	73.20±0.69	85.40±0.39	82.70±0.47	92.93±0.10	91.69±0.14	87.43±0.09
	SumGNN	78.35±0.51	89.05±0.36	87.28±0.08	92.62±0.04	90.80±0.40	85.75±0.10
	SNRI	85.57±0.32	90.15±0.21	88.94±0.36	93.12±0.18	92.64±0.12	87.53±0.11
	KnowDDI	90.06±0.27	93.15±0.19	91.87±0.21	95.05±0.06	93.75±0.05	89.24±0.06
	LaGAT	81.63±0.56	86.21±0.18	85.38±0.23	89.78±0.21	86.33±0.15	83.75±0.36
NAS-based	MR-GNAS	74.24±0.45	88.17±0.24	87.31±0.11	93.85±0.07	91.80±0.03	87.16±0.05
	AutoGEL	76.87±0.63	89.35±0.59	86.14±0.41	94.11±0.32	92.35±0.29	88.13±0.41
	CSSE-DDI-FS	86.31±0.36	91.08±0.21	89.17±0.27	94.35±0.07	93.01±0.06	89.08±0.04
	CSSE-DDI-FF	80.96±0.65	90.27±0.23	88.69±0.31	94.26±0.08	92.74±0.06	88.91±0.09
	CSSE-DDI	**92.08**±0.22	**95.56**±0.15	**94.72**±0.26	**95.47**±0.02	**94.21**±0.05	**89.76**±0.05

4.3 Choices of Search Strategy

To demonstrate the effectiveness of our search strategy, we introduce two variants with different search strategies: (i) **CSSE-DDI w/o MAP**: This variant uses only one trained supernet to serve as a performance evaluator for candidate architectures, instead of generating multiple sub-supernets by Message-Aware Partition (MAP) strategy. (ii) **CSSE-DDI w/o SPOS**: This variant utilizes the message-aware partition strategy to jointly optimize the supernet weights and architectural parameters, without using the Single Path One-Shot (SPOS) strategy [54] .



(a) DrugBank (b) TWOSIDES

Figure 2: Comparison on convergence between the searched architectures by CSSE-DDI and human-designed methods.

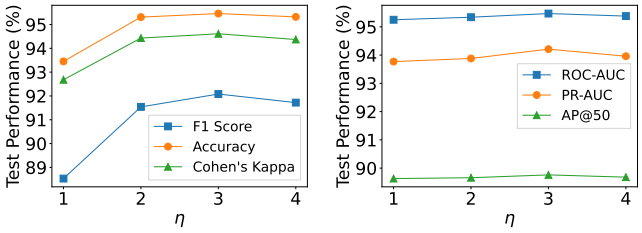
In Table 3, we compare CSSE-DDI with other variants. As can be seen, the absence of either message-aware partition strategy or sampling-based NAS strategy negatively impacts performance. The performance gains achieved through the message-aware partition strategy arise from using multiple sub-supernets, which provide more accurate performance estimations to guide the search process. Regarding the SPOS strategy, it decouples supernet training from architecture search, making it more efficient and robust in practice.

Table 3: Performance of CSSE-DDI using different variants of search algorithm.

Variant	DrugBank	TWOSIDES
CSSE-DDI w/o MAP	90.17±0.29	95.12±0.04
CSSE-DDI w/o SPOS	90.97±0.72	94.89±0.13
CSSE-DDI	**92.08**±0.22	**95.47**±0.02

4.4 Sensitivity Analysis of the Threshold η

Here, we analyze the effect of the threshold η used in subgraph selection space. Figure 3 shows the impact of varying η. As can be observed, model performance continues to get better as the threshold η grows. When the threshold $\eta = 3$, the model performance nears saturation, as larger thresholds do not lead to further improvements. This is likely because most of the essential information for DDI prediction is contained within the 3-hop ego-subgraphs of target drugs. Intuitively, larger subgraphs may provide additional useful information. However, in practice, due to the inherent biases of the search algorithm, achieving an optimal model may be challenging. When η is too large, it may introduce noise and dilute the critical information. A similar phenomenon has been found in the existing work SumGNN [12]. Besides, excessively large thresholds η will only lead to unnecessary expansion of the search space and higher computational costs.



(a) DrugBank (b) TWOSIDES

Figure 3: Performance given different hyperparameter η.

4.5 Performance Comparison in S1 settings

To further validate the effectiveness of our method, we use the S1 setting in the EmerGNN [13] method, to predict drug-drug interactions between emerging drugs and existing drugs. The experimental results are shown in Table 4. A significant performance drop from the transductive setting (S0) to the inductive setting (S1) demonstrates that DDI prediction for new drugs is more challenging. Although Emergnn, which is specifically designed for new drug prediction, achieves optimal performance, CSSE-DDI still demonstrates impressive results, outperforming existing GNN-based and subgraph-based methods. This strong performance is largely due to the robust learning capability of NAS technology in handling unknown data.

Table 4: Experimental results in S1 setting.

Dataset	**Dataset 1: DrugBank**			**Dataset 2: TWOSIDES**		
Task Type	Multi-class			Multi-label		
Methods	F1 Score	Accuracy	Cohen's κ	ROC-AUC	PR-AUC	Accuracy
CompGCN	30.98±3.26	52.76±0.46	37.87±1.28	84.83±1.02	83.68±1.86	74.64±0.79
Decagon	11.39±0.79	32.56±0.92	20.29±1.33	57.49±1.75	59.38±1.09	52.27±1.48
SumGNN	26.57±1.59	44.30±1.04	40.24±1.26	80.02±2.17	78.42±1.62	69.81±1.77
KnowDDI	31.14±1.24	53.44±1.73	43.93±1.17	84.23±2.63	82.58±1.94	74.72±1.51
EmerGNN	**58.13±1.36**	**69.53±1.97**	**62.19±1.62**	87.42±0.39	86.20±0.71	79.23±0.54
CSSE-DDI	37.24±1.13	58.57±0.85	49.97±1.01	**88.33±0.52**	**86.47±0.27**	**80.01±0.39**

4.6 Case Study

4.6.1 Fine-grained Subgraph Selection

We visualize exemplar query-specific subgraphs from the DrugBank dataset in Figure 4, highlighting **domain concepts** such as pharmacokinetics, metabolism, and receptor interactions. As shown, CSSE-DDI can identify distinctive subgraphs containing semantic information to support inference for different queries, revealing pharmacokinetic and metabolic relationships.

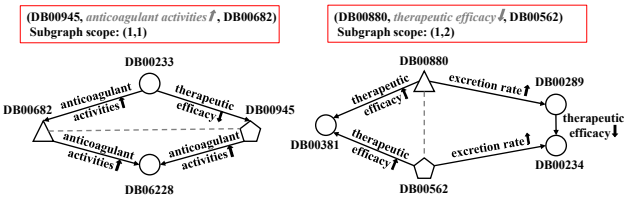


Figure 4: Visualization of the searched subgraphs corresponding to the specific drug pairs.

https://doi.org/10.52202/079017-3478

For example, to predict the interaction between DB00945 (Aspirin) and DB00682 (Warfarin), CSSE-DDI searches out the subgraph scope $(1, 1)$, as depicted on the left part of Figure 4. Firstly, it can be seen from the figure that the therapeutic efficacy of DB00233 (Aminosalicylic acid) can decrease when combined with DB00945 (Aspirin), suggesting similarity between the two drugs [65, 66] Given that DB00233 (Aminosalicylic acid) may increase the anticoagulant activity of DB00682 (Warfarin) and that DB00233 resembles DB00945 (Aspirin), it can be inferred that DB00945 (Aspirin) may similarly increase the anticoagulant activity of DB00682 (Warfarin). This example demonstrates that the identified subgraph contains sufficient semantic information to reason about the interaction between DB00945 (Aspirin) and DB00682 (Warfarin).

4.6.2 Data-specific Encoding Function

Furthermore, we visualize the searched structure of encoding functions across all datasets in Figure 5. It is clearly illustrated that different combinations of the designed operations, i.e., data-specific encoding functions, are obtained.

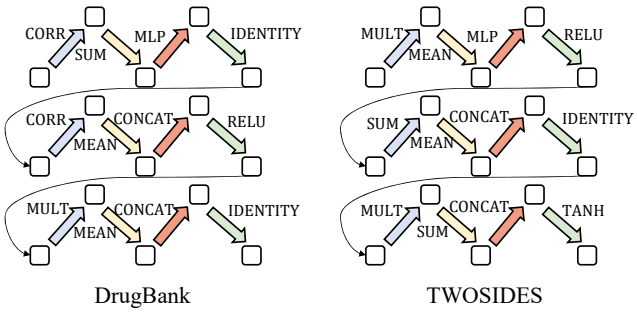


Figure 5: The searched encoding functions on all benchmark datasets.

In particular, the searched message-computing functions contain more `CORR` operations in the DrugBank dataset, while more `MULT` functions are searched in the TWOSIDES dataset. The `CORR` function is non-commutative [67], making it suitable for modeling asymmetric interactions (e.g., metabolic-based interactions) present in DrugBank. While `MULT` is suitable for modeling symmetric relations (phenotype-based interactions) due to its exchangeability [68].

5 Conclusion

We propose a searchable framework, CSSE-DDI, for DDI prediction. Specifically, we design refined search spaces to enable fine-grained subgraph selection and data-specific encoding function optimization. To facilitate efficient search, we introduce a relaxation mechanism to convert the discrete subgraph selection space into a continuous one. Additionally, we employ a subgraph representation approximation strategy to accelerate the search process, addressing the inefficiencies of explicit subgraph sampling. Extensive experiments demonstrate that CSSE-DDI significantly outperforms state-of-the-art methods. Moreover, the search results generated by CSSE-DDI offer interpretability in the context of drug interactions, revealing domain-specific concepts such as pharmacokinetics and metabolism.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Key Research and Development Program of China (Grant No. 2022ZD0160300), in part by the National Science Fund for Distinguished Young Scholars (Grant No. 62025602), in part by the National Natural Science Foundation of China (Grant Nos. U22B2036, 11931015, 62203363, and 92270106), in part by the Technology Innovation Leading Program of Shaanxi (Grant No. 2023GXLH-086), in part by the Beijing Natural Science Foundation (Grant No. 4242039). in part by the Fok Ying-Tong Education Foundation, China (Grant No. 171105), in part by the Fundamental Research Funds for the Central Universities (Grant Nos. G2024WD0151 and D5000240309), and in part by the Tencent Foundation and XPLORER PRIZE.

References

[1] Xuan Lin, Lichang Dai, Yafang Zhou, Zu-Guo Yu, Wen Zhang, Jian-Yu Shi, Dong-Sheng Cao, Li Zeng, Haowen Chen, Bosheng Song, et al. Comprehensive evaluation of deep and graph learning on drug-drug interactions prediction. *arXiv preprint arXiv:2306.05257*, 2023.

[2] Timo Möttönen, Pekka Hannonen, Marjatta Leirisalo-Repo, Martti Nissilä, Hannu Kautiainen, Markku Korpela, Leena Laasonen, Heikki Julkunen, Reijo Luukkainen, Kaisa Vuori, et al. Comparison of combination therapy with single-drug therapy in early rheumatoid arthritis: a randomised trial. *The Lancet*, 353(9164):1568–1573, 1999.

[3] David N Juurlink, Muhammad Mamdani, Alexander Kopp, Andreas Laupacis, and Donald A Redelmeier. Drug-drug interactions among elderly patients hospitalized for drug toxicity. *Jama*, 289(13):1652–1658, 2003.

[4] Bethany Percha and Russ B Altman. Informatics confronts drug–drug interactions. *Trends in pharmacological sciences*, 34(3):178–184, 2013.

[5] Huaqiao Jiang, Yanhua Lin, Weifang Ren, Zhonghong Fang, Yujuan Liu, Xiaofang Tan, Xiaoqun Lv, and Ning Zhang. Adverse drug reactions and correlations with drug–drug interactions: A retrospective study of reports from 2011 to 2020. *Frontiers in Pharmacology*, 13:923939, 2022.

[6] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[7] Santiago Vilar, Eugenio Uriarte, Lourdes Santana, Tal Lorberbaum, George Hripcsak, Carol Friedman, and Nicholas P Tatonetti. Similarity-based modeling in large-scale prediction of drug-drug interactions. *Nature protocols*, 9(9):2147–2163, 2014.

[8] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

[9] Kexin Huang, Cao Xiao, Lucas M Glass, Marinka Zitnik, and Jimeng Sun. Skipgnn: predicting molecular interactions with skip-graph networks. *Scientific reports*, 10(1):21092, 2020.

[10] Mohammad Hussain Al-Rabeah and Amir Lakizadeh. Prediction of drug-drug interaction events using graph neural networks based feature extraction. *Scientific Reports*, 12(1):15590, 2022.

[11] Nguyen Quoc Khanh Le. Predicting emerging drug interactions using gnns. *Nature Computational Science*, 3(12):1007–1008, 2023.

[12] Yue Yu, Kexin Huang, Chao Zhang, Lucas M Glass, Jimeng Sun, and Cao Xiao. Sumgnn: multi-typed drug interaction prediction via efficient knowledge graph summarization. *Bioinformatics*, 37(18):2988–2995, 2021.

[13] Yongqi Zhang, Quanming Yao, Ling Yue, Xian Wu, Ziheng Zhang, Zhenxi Lin, and Yefeng Zheng. Emerging drug interaction prediction enabled by a flow-based graph neural network with biomedical network. *Nature Computational Science*, 3(12):1023–1033, 2023.

[14] Yaqing Wang, Zaifei Yang, and Quanming Yao. Accurate and interpretable drug-drug interaction prediction enabled by knowledge subgraph learning. *Communications Medicine*, 4(1):59, 2024.

[15] Shonosuke Harada, Hirotaka Akita, Masashi Tsubaki, Yukino Baba, Ichigaku Takigawa, Yoshihiro Yamanishi, and Hisashi Kashima. Dual graph convolutional neural network for predicting chemical networks. *BMC bioinformatics*, 21:1–13, 2020.

[16] Mihai Udrescu, Sebastian Mihai Ardelean, and Lucreţia Udrescu. The curse and blessing of abundance—the evolution of drug interaction databases and their impact on drug network analysis. *GigaScience*, 12:giad011, 2023.

[17] Arnold K Nyamabo, Hui Yu, Zun Liu, and Jian-Yu Shi. Drug–drug interaction prediction with learnable size-adaptive molecular substructures. *Briefings in Bioinformatics*, 23(1):bbab441, 2022.

 https://doi.org/10.52202/079017-3478

[18] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[19] Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pages 9448–9457. PMLR, 2020.

[20] Xiaohan Xu, Peng Zhang, Yongquan He, Chengpeng Chao, and Chaoyang Yan. Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs. pages 2341–2347, 2022.

[21] Xin Zheng, Miao Zhang, Chunyang Chen, Chaojie Li, Chuan Zhou, and Shirui Pan. Multi-relational graph neural architecture search with fine-grained message passing. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 783–792. IEEE, 2022.

[22] Zhili Wang, Shimin Di, and Lei Chen. Autogel: An automated graph neural network with explicit link information. *Advances in Neural Information Processing Systems*, 34:24509–24522, 2021.

[23] Yongqi Zhang, Quanming Yao, and James T Kwok. Bilinear scoring function search for knowledge graph learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1458–1473, 2022.

[24] Zhenqian Shen, Yongqi Zhang, Lanning Wei, Huan Zhao, and Quanming Yao. Automated machine learning: From principles to practices. *arXiv e-prints*, pages arXiv–1810, 2018.

[25] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

[26] Hui Zhang, Quanming Yao, James T Kwok, and Xiang Bai. Searching a high performance feature extractor for text recognition network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6231–6246, 2022.

[27] Lanning Wei, Huan Zhao, Zhiqiang He, and Quanming Yao. Neural architecture search for gnn-based graph classification. *ACM Transactions on Information Systems*, 2023.

[28] Sijie Mai, Shuangjia Zheng, Yuedong Yang, and Haifeng Hu. Communicative message passing for inductive relation reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4294–4302, 2021.

[29] Qiaoyu Tan, Xin Zhang, Ninghao Liu, Daochen Zha, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. Bring your own view: Graph neural networks for link prediction with personalized subgraph selection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 625–633, 2023.

[30] Zhen Wang, Haotong Du, Quanming Yao, and Xuelong Li. Search to pass messages for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6160–6172, 2022.

[31] Ziwei Zhang, Xin Wang, and Wenwu Zhu. Automated machine learning on graphs: A survey. pages 4704–4712, 2021.

[32] Jianliang Gao, Zhenpeng Wu, Raeed Al-Sabri, Babatounde Moctard Oloulade, and Jiamin Chen. Autoddi: Drug–drug interaction prediction with automated graph neural network. *IEEE Journal of Biomedical and Health Informatics*, 2024.

[33] Kwei-Herng Lai, Daochen Zha, Kaixiong Zhou, and Xia Hu. Policy-gnn: Aggregation optimization for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 461–471, 2020.

[34] Yaoman Li and Irwin King. Autograph: Automated graph neural network. In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part II 27*, pages 189–201. Springer, 2020.

[35] ZHAO Huan, YAO Quanming, and TU Weiwei. Search to aggregate neighborhood for graph neural network. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 552–563. IEEE, 2021.

[36] Yiyang Zhao, Linnan Wang, Yuandong Tian, Rodrigo Fonseca, and Tian Guo. Few-shot neural architecture search. In *International Conference on Machine Learning*, pages 12707–12718. PMLR, 2021.

[37] Yaqing Wang, Zaifei Yang, and Quanming Yao. Accurate and interpretable drug-drug interaction prediction enabled by knowledge subgraph learning. *arXiv preprint arXiv:2311.15056*, 2023.

[38] Shimin Di and Lei Chen. Message function search for knowledge graph embedding. In *Proceedings of the ACM Web Conference 2023*, pages 2633–2644, 2023.

[39] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[40] Jiaxuan You, Zhitao Ying, and Jure Leskovec. Design space for graph neural networks. *Advances in Neural Information Processing Systems*, 33:17009–17021, 2020.

[41] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33:13260–13271, 2020.

[42] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

[43] 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.

[44] Shimin Di, Quanming Yao, Yongqi Zhang, and Lei Chen. Efficient relation-aware scoring function search for knowledge graph embedding. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1104–1115. IEEE, 2021.

[45] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[46] Xuanyi Dong, David Jacob Kedziora, Katarzyna Musial, Bogdan Gabrys, et al. Automated deep learning: Neural architecture search is not the end. *Foundations and Trends® in Machine Learning*, 17(5):767–920, 2024.

[47] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. In *International Conference on Learning Representations*, 2019.

[48] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[49] Yongqi Zhang, Zhanke Zhou, Quanming Yao, Xiaowen Chu, and Bo Han. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3446–3457, 2023.

[50] Yongqi Zhang and Quanming Yao. Knowledge graph reasoning with relational digraph. In *Proceedings of the ACM web conference 2022*, pages 912–924, 2022.

[51] Zhanke Zhou, Yongqi Zhang, Jiangchao Yao, Quanming Yao, and Bo Han. Less is more: One-shot subgraph reasoning on large-scale knowledge graphs. 2024.

[52] Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*, pages 3469–3489. PMLR, 2022.

 https://doi.org/10.52202/079017-3478

[53] Zizhao Zhang, Xin Wang, Chaoyu Guan, Ziwei Zhang, Haoyang Li, and Wenwu Zhu. Autogt: Automated graph transformer architecture search. In *International Conference on Learning Representations*, 2023.

[54] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 544–560. Springer, 2020.

[55] Andrew Brock, Theo Lim, JM Ritchie, and Nick Weston. Smash: One-shot model architecture search through hypernetworks. In *International Conference on Learning Representations*, 2018.

[56] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018.

[57] Kaicheng Yu, Christian Sciuto, Martin Jaggi, Claudiu Musat, and Mathieu Salzmann. Evaluating the search phase of neural architecture search. In *International Conference on Learning Representations*, 2020.

[58] Zhaoxuan Tan, Zilong Chen, Shangbin Feng, Qingyue Zhang, Qinghua Zheng, Jundong Li, and Minnan Luo. Kracl: Contrastive learning with graph context modeling for sparse knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pages 2548–2559, 2023.

[59] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 201.

[60] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, 2020.

[61] Hui Yu, KangKang Li, WenMin Dong, ShuangHong Song, Chen Gao, and JianYu Shi. Attention-based cross domain graph neural network for prediction of drug–drug interactions. *Briefings in Bioinformatics*, 24(4):bbad155, 2023.

[62] Junkai Cheng, Yijia Zhang, Hengyi Zhang, Shaoxiong Ji, and Mingyu Lu. Transfol: A logical query model for complex relational reasoning in drug-drug interaction. *IEEE Journal of Biomedical and Health Informatics*, 2024.

[63] Yue Hong, Pengyu Luo, Shuting Jin, and Xiangrong Liu. Lagat: link-aware graph attention network for drug–drug interaction prediction. *Bioinformatics*, 38(24):5406–5412, 2022.

[64] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[65] EJ Ariëns. Reduction of drug action by drug combination. In *Progress in Drug Research/Fortschritte der Arzneimittelforschung/Progrès des recherches pharmaceutiques*, pages 11–58. Springer, 1970.

[66] Julie Foucquier and Mickael Guedj. Analysis of drug combinations: current methodological landscape. *Pharmacology research & perspectives*, 3(3):e00149, 2015.

[67] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[68] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, 2015.

[69] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

[70] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[71] Yongqi Zhang, Quanming Yao, and Lei Chen. Interstellar: Searching recurrent architecture for knowledge graph embedding. *Advances in Neural Information Processing Systems*, 33: 10030–10040, 2020.

[72] Youhei Akimoto, Shinichi Shirakawa, Nozomu Yoshinari, Kento Uchida, Shota Saito, and Kouhei Nishida. Adaptive stochastic natural gradient method for one-shot neural architecture search. In *International Conference on Machine Learning*, pages 171–180. PMLR, 2019.

 https://doi.org/10.52202/079017-3478

Appendix

https://doi.org/10.52202/079017-3478

A More Method Details

A.1 Subgraph Encoding Space

An expressive subgraph encoding space can be naturally designed by including human-designed operations, the details of which are given in Table 5.

Table 5: The operations used in our search space.

Function name	Operations
Message Computing Function	SUB, MULT, CORR, ROTATE
Aggregation Function	SUM, MAX,MEAN
Combination Function	MLP, CONCAT
Activation Function	RELU, TANH, IDENTITY

In particular, given the embedding $\mathbf{h}_u$ of node u and the embedding $\mathbf{h}_r$ of interaction r, the message computing function takes the following form: $\texttt{MES}_{\texttt{SUB}} = \mathbf{h}_u - \mathbf{h}_r$, $\texttt{MES}_{\texttt{MULT}} = \mathbf{h}_u * \mathbf{h}_r$, $\texttt{MES}_{\texttt{CORR}} = \mathbf{h}_u \star \mathbf{h}_r$, $\texttt{MES}_{\texttt{ROTATE}} = \mathbf{h}_u \circ \mathbf{h}_r$, where $\star$ stands for the circular correlation operation [67], $\circ$ represents the rotation operation [69].

A.2 Robust Search Algorithm

We adopt the single path one-shot (SPOS) training strategy to solve the customized search problem, which decouple supernet training and architecture searching. In particular, definition 1 can be transformed into a two-step optimaztion [54]:

$$\arg\max_{\boldsymbol{\alpha}\in\mathcal{A},\mathcal{G}_{u,v}\in\mathcal{S}_{u,v}} \sum_{(u,r,v)\in\mathcal{D}_{\text{val}}} \mathcal{M}(\mathbf{W}^*;\mathcal{G}_{u,v};\boldsymbol{\alpha}), \tag{8}$$

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathbb{E}_{\boldsymbol{\alpha}\in\mathcal{A}} \sum_{(u,r,v)\in\mathcal{D}_{\text{tra}}} \mathcal{L}(\mathbf{W};\mathcal{G}_{u,v};\boldsymbol{\alpha}), \tag{9}$$

where $\mathbf{W}$ denotes the shared learnable weights in the supernet with its optimal value $\mathbf{W}^*$ for all the architectures in the overall search space.

Eq. (9),(8) represent the supernet training and architecture searching phase, respectively. In the following, we will describe the detailed process of the two phases.

A.2.1 Supernet Training

In supernet training phase, a sub-model $\boldsymbol{\alpha}$ is sampled according to the discrete distribution $\pi(\mathcal{A})$. Thus, Eq. (9) can be formulated as

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathbb{E}_{\boldsymbol{\alpha}\sim\pi(\mathcal{A})} \sum_{(u,r,v)\in\mathcal{D}_{\text{tra}}} \mathcal{L}(\mathbf{W};\mathcal{G}_{u,v};\boldsymbol{\alpha}), \tag{10}$$

where the discrete distribution $\pi(\mathcal{A})$ is set to uniform distribution.

First, we need to perform single path sampling to train the supernet until it converges. In the next step, we need to partition the supernet into sub-supernets. which is a key step aiming to isolate operations that are coupled with each other. This allows the supernet to be trained and converge more stably.

In our supernet, we use a message-aware partitions strategy due to the fact that the degree of dissimilarity between the operations in the message computing function MES is much higher compared with others. These operations focus on capture different semantic types of interactions, which has been discussed in existing works [70, 69, 58]. Therefore, we partition four operations of the message computing function of the first layer of the supernet, to improve the accuracy of the performance estimation.

After partitioning operation, we initialize four sub-supernets with weights transferred from the original supernet. Next, we train these sub-supernets to convergence by sampling single path. Here, the supernet training phase is all done.

https://doi.org/10.52202/079017-3478

A.2.2 Architecture Searching

After completing sub-supernet training phase, we have obtained well-trained supernet weights. In the searching phase, Eq. (8) can be transformed as

$$\arg\max_{\mathcal{G}_{u,v}\in\mathcal{S}_{u,v}} \sum_{(u,r,v)\in\mathcal{D}_{\text{val}}} \mathcal{M}(\mathbf{W}^*; \mathcal{G}_{u,v}; \boldsymbol{\alpha}), \tag{11}$$

$$\text{s.t. } \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \sum_{(u,r,v)\in\mathcal{D}_{\text{val}}} \mathcal{M}(\mathbf{W}^*; \mathcal{G}_{u,v}; \boldsymbol{\alpha}), \tag{12}$$

For suugraph encoding function searching in Eq. (12), following [71], we adopt stochastic relaxation on $\boldsymbol{\alpha}$ and natural policy gradient strategy [72] to obtain the optimal subgraph encoding function $\boldsymbol{\alpha}^*$. For subgraph selection in Eq. (11), we obtain the optimal subgraph $\mathcal{G}^*_{u,v}$ by preserving the subgraph with the largest probability $p^{i,j}_{u,v}$, i.e.,

$$\mathbf{z}^{i,j}_{u,v} = f(\mathcal{G}^{i,j}_{u,v}), \tag{13}$$

$$\beta^{i,j}_{u,v} = g(\mathbf{z}^{i,j}_{u,v}), \tag{14}$$

$$p^{i,j}_{u,v} = \frac{\exp(\log(\beta^{i,j}_{u,v} + \mathbf{G}_{i,j})/\tau)}{\sum_{i',j'=1}^{\eta} \exp(\log(\beta^{i',j'}_{u,v} + \mathbf{G}_{i',j'})/\tau)}, \tag{15}$$

$$\mathcal{G}^*_{u,v} = \arg\max_{\mathcal{G}^{i,j}_{u,v}} p^{i,j}_{u,v}(\mathcal{G}^{i,j}_{u,v} \in S_{u,v}). \tag{16}$$

B More Experiment Setting

B.1 Datasets

Experiments are performed on two public benchmark DDI datasets: DrugBank and TWOSIDES.

DrugBank DrugBank dataset contains 1,710 drugs and drug pairs, which are related to 86 types of pharmacological interactions between drugs, such as increase of anticoagulant activity, decrease of excretion rate and etc.

TWOSIDES TWOSIDES dataset contains 604 drugs and drug pairs with 200 drug side effects as interaction labels. For each edge, it may be associated with multiple interactions.

The detailed descriptions for datasets are presented in Table 6 and Table 7.

Table 6: The statistics of the datasets.

Dataset	#nodes	#edges	#interaction types
DrugBank	1,710	134641	86
TWOSIDES	604	57778	200

Table 7: Diverse semantic properties in drug-drug interactions.

Dataset	Interaction Type	Examples	Semantic Property
DrugBank	Metabolic levels-based	#Drug1 may decrease the excretion rate of #Drug2	asymmetry $(r(x,y) \nRightarrow r(y,x))$
TWOSIDES	Phenotype-based	Combination of #Drug 1 and #Drug 2 may cause headaches	symmetry $(r(x,y) \Rightarrow r(y,x))$

B.2 Evaluation Metric

We follow [12] to evaluate our method. Specifically, in terms of the multi-class prediction on DrugBank, we followc[12] and evaluate the performance by three metrics: (i) Macro F1 score (Macro F1) is computed by taking the arithmetic mean (aka unweighted mean) of all the per-class F1 scores. (ii) Accuracy (ACC) is calculated by dividing the number of correct predictions by the total prediction

number. (iii) Coken's Kappa (Cohen's κ) measures inter-rater reliability. As to the multi-label prediction on TWOSIDES, we consider the following measure and use the average performance over all interaction types: (i) ROC-AUC (AUROC) stands for "Area Under the Curve (AUC)" of the "Receiver Operating Characteristic (ROC)" curve. (ii) PR-AUC (AUPRC) is the average area under precision-recall curve. (iii) AP@50 is the average precision at 50.

B.3 Implementation and Hyperparameters

All the experiments are implemented in Python with the PyTorch framework [64] and run on a server machine with single NVIDIA RTX 3090 GPU with 24GB memory and 64GB of RAM. Our code is added in the supplementary material.

For CSSE-DDI, we set the epoch to 400 for training supernet and set the epoch to 400 for training sub-supernets. We set the the temperature parameter as 0.05. Repeat 5 times with different seeds, we can get 5 candidates. The searched candidates are finetuned individually with the hyper-parameters. In the stage of fine-tuning, we use the ReduceLROnPlateau scheduler to adjust the learning rate dynamically. Each candidate has 10 hyper steps. In each hyper step, a set of hyperparameter will be sampled from Table 8.

Table 8: Hyperparameters we used during the fine-tuning stage.

Hyperparameter	**Value range**
Learning rate	$[10^{-3.1}, 10^{-2.9}]$
Weight decay	$[10^{-5}, 10^{-3}]$

C More Experimental Results

C.1 Subgraph Scope Distribution Analysis

We visualize the learned distributions of subgraph scope on all datasets by using CSSE-DDI in Figure 6. By comparing the distributions across different benchmarks, we have the following observation: CSSE-DDI can effectively learn different subgraph scope distributions for various datasets. By identifing specific subgraph scopes for different queries, CSSE-DDI is able to precisely control the extent of information propagation required for reasoning about the interactions of different drug pairs. In addition, our method can skip some subgraph scopes if they are not optimal for any queries. For example, no queries are assigned to the propagation scope $(3, 3)$ on TWOSIDES dataset. It is worth mentioning that our searched subgraph scopes are consistent with the sensitivity analysis results for the hop of subgraph in SumGNN [12], which further validates the effectiveness of our approach.

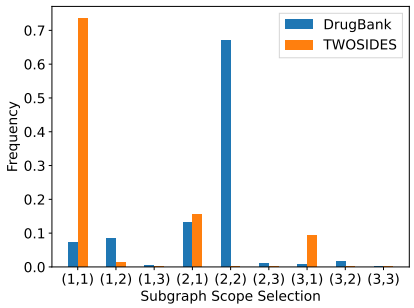


Figure 6: Distribution of the searched subgraph scopes by CSSE-DDI on all benchmark datasets.

https://doi.org/10.52202/079017-3478

D Some Discussions about Checklist

D.1 Limitations

There are three limitations for CSSE-DDI. (1) CSSE-DDI is focused on method design rather than system design. In the future, we will co-design the algorithm and the system to further improve the efficiency. (2) At present, CSSE-DDI only search for data-specific components of subgraph-based pipeline, while hyper-parameters are also important for DDI prediction. A promising direction is to explore how to efficiently search network architectures and hyper-parameters simultaneously.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in our paper and important assumptions and limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

https://doi.org/10.52202/079017-3478

Justification: We discuss the limitations in Section D.1 of the Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the complete code that runs correctly and hyperparameter configurations in the supplemental material and Section B.3 to ensure reproducibility and transparency.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the datasets, the complete code that runs correctly and hyperparameter configurations in the supplemental material and Section B.3 to ensure reproducibility and transparency.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

 https://doi.org/10.52202/079017-3478

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the data splits, hyperparameter configurations, and other experimental details in the supplemental material and Section B.3 to ensure reproducibility and transparency.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All of the methods are run for five times on the different random seeds with mean value and standard deviation reported on the testing data, as shown in Table 2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the configuration of running environment in the supplemental material and Section B.3 to ensure reproducibility and transparency.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We would claim that this work does not raise any ethical concerns. Besides, this work does not involve any human subjects, practices to data set releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We believe that this work is expected to have a positive impact in the field of health care and medicine, and by predicting drug-drug interactions, the method has a positive effect in reducing experimental costs and assisting in the prediction of drug-drug interactions.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

https://doi.org/10.52202/079017-3478

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The assets we submitted have detailed documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3478