MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **96** (2026) 5–41
ISSN: 0340–6253
doi: 10.46793/match.96-1.25325

A Multi-Scale Graph Neural Process with Cross-Drug Co-Attention for Drug-Drug Interactions Prediction

Zimo Yan, Jie Zhang, Zheng Xie*, Song Yiping, Li Hao

National University of Defense Technology

yanzimo20@nudt.edu.cn, zhangjie@nudt.edu.cn, xiezheng81@nudt.edu.cn, songyiping@nudt.edu.cn, lihao22@nudt.edu.cn

(Received September 29, 2025)

Abstract

Predicting drug-drug interactions (DDIs) is a critical challenge in medication safety and drug development. Existing methods, however, often fail to effectively capture the full spectrum of structural information, from local functional groups to global molecular topology, and typically lack principled mechanisms to quantify prediction confidence. To address these limitations, we propose the Multi-scale Graph Neural Process for DDI (MPNP-DDI), a novel framework that employs an iterative message-passing scheme to build a hierarchy of graph representations. These multi-scale features are then dynamically fused by a cross-drug co-attention mechanism to generate context-aware embeddings for interacting drug pairs. By providing accurate, generalizable, and uncertainty-aware predictions built upon multi-scale structural features, MPNP-DDI represents a reliable computational tool for pharmacovigilance, polypharmacy risk assessment, and precision medicine.

*Corresponding author.

1 Introduction

The concurrent use of multiple medications, known as polypharmacy, is increasingly common, elevating the risk of adverse drug events stemming from unforeseen drug-drug interactions [1]. To mitigate these risks, predicting these interactions serves as a cornerstone of pharmacovigilance and clinical decision support. However, the challenge is immense, as the number of potential DDIs grows combinatorially with the number of available drugs, making exhaustive experimental screening infeasible [2]. This reality underscores the critical importance of developing accurate and scalable computational models to forecast DDI risks preemptively.

Initial computational approaches for DDI prediction relied heavily on literature mining to extract known interactions from biomedical texts [3], or similarity-based methods that assume drugs with similar properties (e.g., chemical structure, target proteins) are likely to share similar interaction profiles [4]. In recent years, Graph Neural Networks (GNNs) have emerged as the state-of-the-art for learning from molecular data [5]. When applying GNNs to the DDI problem, which inherently involves a pair of drugs, the dual-GNN architecture has become a common paradigm [6]. In this setup, two separate GNNs process the paired drugs independently, and their final embeddings are concatenated for prediction [7, 8].

Beyond these foundational models, emerging strategies are tackling the DDI prediction problem with greater complexity. Knowledge graph-based methods embed drugs within a larger biomedical network, incorporating heterogeneous information such as proteins, diseases, and side effects to enrich drug representations [9, 10]. In parallel, multi-modal approaches aim to fuse diverse data sources, such as molecular structures and textual descriptions, to create more comprehensive drug profiles [11]. Other advanced models have begun to incorporate co-attention mechanisms to model substructure-level interactions [12].

Despite these advancements, a fundamental limitation persists, as many models are built upon standard GNNs that operate at a single, fixed analytical scale. This prevents them from simultaneously capturing fine-grained local substructures and global molecular topology, often seeing the

trees but not the forest [13], while also lacking a mechanism to dynamically focus on the most salient chemical motifs [14]. This architectural scale-insensitivity leads to a more profound conceptual flaw: the generation of static, context-agnostic drug representations. In prevalent dual-GNN pipelines, the representation of Drug A is computed in an "information silo," entirely independent of its partner, Drug B [7, 8]. This approach is fundamentally misaligned with chemical reality, where a drug's interactive potential is dynamic and context-dependent. A truly effective model must therefore first perceive features across multiple scales to then generate a dynamic, context-aware representation that reflects how these features are expressed in the presence of a specific partner [12].

Motivation: This raises the following question: *Can we design a DDI prediction model that learns dynamic, context-aware representations from a rich hierarchy of multi-scale structural features, while also quantifying its reliability for high-stakes clinical predictions?* To address this challenge, we introduce the Multi-scale Graph Neural Process for DDI (MPNP-DDI), as illustrated in Figure 1. We select the Graph Neural Process framework for its unique ability to learn a distribution over functions, enabling robust generalization to entirely new molecules not seen during training. This interactive process yields a context-aware representation for predicting both the DDI event and the model's uncertainty.

Primary contributions. Our primary contributions are threefold:

1. Multi-Scale Representation Learning: We use stacked Graph Neural Process blocks, operating on both the original molecular graph and its line graph (to model bond-level interactions), to build a hierarchy of stochastic representations from local motifs to global topology.

2. Context-Aware Feature Fusion: A cross-drug co-attention mechanism dynamically fuses these multi-scale features, breaking the information silo by generating a unique, context-dependent embedding for each interacting drug pair.

3. Principled Uncertainty & Generalization: By learning a distribution over functions, the GNP framework not only provides principled uncertainty estimates but also exhibits strong generalization to unseen

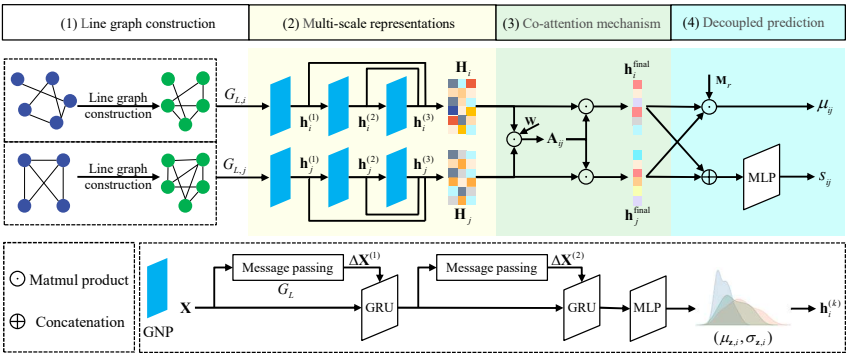


Figure 1. The architecture of MPNP-DDI. Stacked GNP blocks generate multi-scale representations for each drug, which are then fused by a cross-drug co-attention mechanism to enable context-aware, probabilistic DDI prediction.

molecules, which is a critical feature for real-world drug discovery that distinguishes it from many standard GNNs.

2 Literature review

This section reviews the evolution of computational DDI prediction, structured to mirror the methodological categories outlined in the introduction.

Foundational and Similarity-Based Methods. Initial computational strategies were built on two pillars. Literature mining employed Natural Language Processing (NLP) to extract known DDIs from biomedical texts [3, 15], a method inherently unable to predict novel interactions. In parallel, similarity-based methods operated on the principle that similar drugs exhibit similar behaviors [4]. These models used feature vectors like chemical fingerprints or target profiles [16, 17] to infer interactions. Their main limitation is a reliance on hand-crafted features and the "similarity assumption," which may not always hold true.

Graph Neural Network-Based Prediction. The advent of Graph Neural Networks (GNNs) marked a paradigm shift, enabling models to learn representations directly from the molecular graph [5]. This led to the prevalence of the **dual-GNN architecture**, where two GNNs (or a

single shared-weight GNN) independently process the paired drugs to generate fixed-size embeddings. These embeddings are then concatenated and fed into a classifier to predict the interaction type [7,8]. This approach became the new standard but established the "information silo" problem, as drug representations are computed without context from their interacting partner.

Knowledge Integration and Architectural Enhancements. To move beyond the standard dual-GNN, two advanced strategies emerged. The first involves enriching drug representations with external data, using Knowledge Graphs (KGs) [9, 10, 18] or multi-modal approaches [11, 19]. The second strategy focuses on improving the GNN itself by enhancing its ability to capture salient structural information. For instance, some works focus on enhancing structural feature extraction directly [20] or identifying key chemical motifs, a concept also central to related tasks like molecular design [21]. In the DDI context, models like SSI-DDI [22], GMPNN-CS [23], and DGNN-DDI [12] incorporated attention or gated mechanisms to focus on important substructures. However, even in these models, co-attention is often applied as a late-stage fusion step on precomputed, static features, failing to fully model the dynamic nature of drug interactions.

Probabilistic and Context-Aware Modeling. The limitations of static models motivate exploring more advanced frameworks. Graph Neural Processes (GNPs) represent a promising frontier [24]. Unlike deterministic GNNs, GNPs learn a distribution over functions on graphs. This probabilistic nature is inherently suited for few-shot generalization to new drugs [25] and, most critically, provides a principled mechanism for uncertainty quantification. The vast majority of DDI models lack this feature, a significant shortcoming in clinical settings where reliability is paramount. A model that can express its own uncertainty would be transformative, guiding both clinical decisions and future research. Despite this potential, applying GNPs to DDI prediction remains largely unexplored, highlighting a critical gap this work aims to address.

3 Preliminaries

This section introduces the fundamental graph-based representations for drugs and formally defines the task of multi-label, probabilistic DDI prediction.

3.1 Graph representation and task formulation

A drug molecule is represented as a graph $G = (V, E)$, where V is the set of atoms (nodes) and E is the set of chemical bonds (edges). Each atom $v \in V$ is associated with an initial feature vector $\mathbf{x}_v \in \mathbb{R}^{d_v}$, and each bond $e_{uv} \in E$ has a feature vector $\mathbf{e}_{uv} \in \mathbb{R}^{d_e}$. These features are projected into a unified hidden dimension d_h to yield initial states $\mathbf{x}_v^{(0)}$ and $\mathbf{e}_{uv}^{(0)}$.

To capture a richer structural context, we model molecules from two complementary viewpoints: atom-level interactions, represented by the standard graph G, and bond-level interactions. For the latter, we construct the **line graph** $G_L = (V_L, E_L)$. In G_L, each node corresponds to a bond in G, and an edge exists between two nodes if their corresponding bonds in G share a common atom. Operating on both G and G_L provides the structural foundation for our multi-scale feature extraction.

We formulate DDI prediction as a **multi-label classification** problem over a set of R predefined DDI types. Given a dataset of drug pairs $\mathcal{D} = \{(G_i, G_j, \mathbf{y}_{ij})\}$, the label $\mathbf{y}_{ij} \in \{0, 1\}^R$ is a binary vector where $y_{ij,r} = 1$ if drug i and drug j exhibit the r-th type of interaction, and 0 otherwise.

3.2 Probabilistic modeling objective

Our goal is to learn a probabilistic model f_θ that maps a new, potentially unseen drug pair (G_i, G_j) to a distribution over the R possible interaction types. Specifically, the model outputs a tuple $(\boldsymbol{\mu}_{ij}, \mathbf{s}_{ij})$, where $\boldsymbol{\mu}_{ij} \in \mathbb{R}^R$ is a vector of logits and $\mathbf{s}_{ij} \in \mathbb{R}^R$ is a vector of log-variances:

$$f_\theta : (G_i, G_j) \mapsto (\boldsymbol{\mu}_{ij}, \mathbf{s}_{ij}) \tag{1}$$

Table 1. Summary of key notations.

Notation	Description
$\mathcal{D}_{\text{train}}$	Training set of labeled drug pairs.
G_i, G_j	Molecular graphs for a drug pair.
R	Number of distinct DDI relation types.
$\mathbf{y}_{ij} \in \{0,1\}^R$	Multi-label interaction vector.
$\mathbf{x}_v, \mathbf{e}_{uv}$	Raw atom (node) and bond (edge) features.
d_h	Hidden dimension for all embeddings.
$\mathbf{x}_v^{(0)}, \mathbf{e}_{uv}^{(0)}$	Initial hidden states for atoms and bonds.
$G_{L,i}$	Line graph derived from G_i.
$\mathcal{H}_i$	Set of multi-scale embeddings $\{\mathbf{h}_i^{(k)}\}_{k=1}^K$ for a drug.
$f_\theta(\cdot,\cdot)$	The DDI prediction model with parameters θ.
$(\boldsymbol{\mu}_{ij}, \mathbf{s}_{ij})$	Model output: interaction logits and log-variances (vectors of size R). $\sigma_{ij,r}^2 = \exp(s_{ij,r})$.
$\mathcal{L}_{\text{MPNP}}$	The total loss function.
$\mathcal{L}_{\text{pred}}, \mathcal{L}_{\text{unc}}, \mathcal{L}_{\text{kl}}$	Prediction, uncertainty, and KL loss components.
$\lambda_{\text{unc}}, \lambda_{\text{kl}}$	Weights for the loss components.

For each interaction type $r \in \{1, \ldots, R\}$, the model provides an interaction probability $p_{ij,r} = \text{sigmoid}(\mu_{ij,r})$ and an estimated uncertainty, captured by the variance $\sigma_{ij,r}^2 = \exp(s_{ij,r})$. This fine-grained, per-relation uncertainty is critical for assessing prediction reliability before informing decisions. The model is trained by optimizing a composite objective function, which will be detailed in the Methodology section.

4 Methodology

We propose the Multi-scale Graph Neural Process for DDI (MPNP-DDI), a framework designed for multi-label, probabilistic DDI prediction. The architecture comprises four key stages: (1) a dual message-passing scheme operating on both atom and bond graphs; (2) a hierarchical Graph Neural Process encoder for multi-scale stochastic representation learning; (3) a cross-drug co-attention mechanism for context-aware feature fusion; and (4) a decoupled prediction head for multi-label probabilistic output.

4.1 Hierarchical structure encoder

To capture both local atomic environments and higher-order bond interactions, we employ a dual message-passing scheme that iterates between the molecular graph G_i and its line graph $G_{L,i}$. First, raw atom and bond features are projected into a unified hidden dimension d_h:

$$\mathbf{x}_v^{(0)} = \mathrm{PReLU}(\mathrm{BatchNorm}(\phi_v(\mathbf{x}_v))), \quad \mathbf{e}_{uv}^{(0)} = \phi_e(\mathbf{e}_{uv}) \tag{2}$$

Within each of the K GNP blocks, representations are refined over T iterations. The process for each iteration t begins on the line graph $G_{L,i}$, where bond representations are updated by passing messages between adjacent bonds. This bond-to-bond communication is defined as:

$$\mathbf{e}_{uv}^{(t+1)} = \mathrm{Update}_e(\mathbf{e}_{uv}^{(t)}, \mathrm{Aggregate}_e(\{\mathbf{m}_{wv\to uv}^{(t)} \mid e_{wv} \in \mathcal{N}_{G_L}(e_{uv})\})) \tag{3}$$

where Update_e and $\mathrm{Aggregate}_e$ are learnable functions (e.g., GRU and summation). The key to our dual scheme is that these newly refined bond representations immediately serve as messages for the second stage, which occurs on the original graph G_i. Here, each atom aggregates the updated states from its incident bonds:

$$\Delta\mathbf{x}_v^{(t)} = \mathrm{Aggregate}_v(\{\mathbf{e}_{uv}^{(t+1)} \mid u \in \mathcal{N}_G(v)\}) \tag{4}$$

Finally, a Gated Recurrent Unit (GRU) [27] integrates this aggregated signal to update the atom's hidden state, enabling stable long-range information propagation:

$$\mathbf{x}_v^{(t+1)} = \mathrm{GRU}(\Delta\mathbf{x}_v^{(t)}, \mathbf{x}_v^{(t)}) \tag{5}$$

This two-stage iterative refinement allows the model to learn complex structural motifs by explicitly modeling the flow of information from bonds to adjacent bonds, and then from bonds back to atoms.

4.2 Multi-scale stochastic representation learning

After T message-passing iterations within block k, we generate a stochastic graph-level representation.

Stochastic Graph Readout. The final node representations $\{\mathbf{x}_v^{(T)}\}_{v\in V_i}$ are pooled to parameterize a diagonal Gaussian posterior distribution $q(\mathbf{z}_i^{(k)}|G_i)$. We use global mean pooling followed by two separate MLPs to produce the mean $\boldsymbol{\mu}_{\mathbf{z},i}^{(k)}$ and log-variance $\log(\boldsymbol{\sigma}_{\mathbf{z},i}^{(k)})^2$:

$$\bar{\mathbf{x}}_i = \text{Mean}(\{\mathbf{x}_v^{(T)} \mid v \in V_i\}) \tag{6}$$

$$\boldsymbol{\mu}_{\mathbf{z},i}^{(k)} = \text{MLP}_{\mu}^{(k)}(\bar{\mathbf{x}}_i), \quad \log(\boldsymbol{\sigma}_{\mathbf{z},i}^{(k)})^2 = \text{MLP}_{\sigma}^{(k)}(\bar{\mathbf{x}}_i) \tag{7}$$

A stochastic graph representation $\mathbf{h}_i^{(k)}$ is then sampled using the reparameterization trick:

$$\mathbf{h}_i^{(k)} = \boldsymbol{\mu}_{\mathbf{z},i}^{(k)} + \boldsymbol{\sigma}_{\mathbf{z},i}^{(k)} \odot \boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{8}$$

By stacking K such blocks, we obtain a set of multi-scale representations for drug d_i, denoted as $\mathcal{H}_i = \{\mathbf{h}_i^{(1)}, \ldots, \mathbf{h}_i^{(K)}\}$.

4.3 Dynamic Interaction Modeling with Co-Attention

To break the "information silo" of independent drug encoding, a co-attention mechanism dynamically fuses the multi-scale representations $\mathcal{H}_i$ and $\mathcal{H}_j$ from an interacting pair. First, a cross-scale affinity matrix $\mathbf{A}_{ij} \in \mathbb{R}^{K\times K}$ is computed:

$$(A_{ij})_{kl} = (\mathbf{h}_i^{(k)})^\top \mathbf{W} \mathbf{h}_j^{(l)} \tag{9}$$

where $\mathbf{W}$ is a learnable parameter matrix. Aggregated importance scores for each scale are obtained by summing over the rows and columns of $\mathbf{A}_{ij}$, which are then normalized via softmax to yield attention weights $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$. The final context-aware embeddings are computed as a weighted sum:

$$\mathbf{h}_i^{\text{final}} = \sum_{k=1}^{K} \alpha_{ik} \mathbf{h}_i^{(k)}, \quad \mathbf{h}_j^{\text{final}} = \sum_{l=1}^{K} \alpha_{jl} \mathbf{h}_j^{(l)} \tag{10}$$

4.4 Multi-label probabilistic prediction head

The final stage uses a decoupled architecture to predict interaction probabilities and their associated uncertainties for all R DDI types.

Prediction Head. To model the rich, multi-relational nature of DDIs, we compute a logit $\mu_{ij,r}$ for each relation type r using the RESCAL model [26]. This allows the model to learn a unique interaction pattern for each DDI type:

$$\mu_{ij,r} = (\mathbf{h}_i^{\text{final}})^\top \mathbf{M}_r \mathbf{h}_j^{\text{final}} \tag{11}$$

where $\mathbf{M}_r \in \mathbb{R}^{d_h \times d_h}$ is a learnable, relation-specific scoring matrix. The final output is a logit vector $\boldsymbol{\mu}_{ij} \in \mathbb{R}^R$.

Uncertainty Head. A separate MLP operates on the concatenated final embeddings to predict a log-variance $s_{ij,r}$ for each relation type, yielding an uncertainty vector $\mathbf{s}_{ij} \in \mathbb{R}^R$:

$$\mathbf{s}_{ij} = \text{MLP}_{\text{unc}}([\mathbf{h}_i^{\text{final}}; \mathbf{h}_j^{\text{final}}]) \tag{12}$$

4.5 Training objective

The model is trained end-to-end by minimizing a composite loss function $\mathcal{L}_{\text{MPNP}}$, which balances predictive accuracy, uncertainty calibration, and latent space regularization:

$$\mathcal{L}_{\text{MPNP}} = \mathcal{L}_{\text{pred}} + \lambda_{\text{unc}}\mathcal{L}_{\text{unc}} + \lambda_{\text{kl}}\mathcal{L}_{\text{kl}} \tag{13}$$

where λ_{unc} and λ_{kl} are hyperparameters that control the weight of each component.

The first component is the Prediction Loss ($\mathcal{L}_{\text{pred}}$), which is the standard binary cross-entropy with logits loss. It is applied independently to each of the R interaction types and averaged over the batch:

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\left[\frac{1}{R}\sum_{r=1}^{R}(-y_{ij,r}\mu_{ij,r} + \log(1 + \exp(\mu_{ij,r})))\right] \tag{14}$$

The second component, the Uncertainty Loss ($\mathcal{L}_{\text{unc}}$), acts as a regularization term. It encourages the model to assign higher variance (i.e., lower

confidence) to incorrect predictions for each relation type:

$$\mathcal{L}_{\text{unc}} = \mathbb{E}\left[\frac{1}{R}\sum_{r=1}^{R}\left((\text{sigmoid}(\mu_{ij,r}) - y_{ij,r})^2 \exp(-s_{ij,r}) + s_{ij,r}\right)\right] \tag{15}$$

Finally, the KL Regularization ($\mathcal{L}_{\text{kl}}$) term is the Kullback-Leibler divergence between the learned posterior distribution $q(\mathbf{z}_i^{(k)}|G_i)$ and a standard normal prior $\mathcal{N}(0, \mathbf{I})$. This divergence is summed over all scales and both drugs in a given pair:

$$\mathcal{L}_{\text{kl}} = \mathbb{E}\left[\sum_{d\in\{i,j\}}\sum_{k=1}^{K}\text{KL}[q(\mathbf{z}_d^{(k)}|G_d)||\mathcal{N}(0, \mathbf{I})]\right] \tag{16}$$

5 Theoretical analysis

For DDI prediction models intended for clinical use, formal guarantees on training stability and on generalization to novel drugs are essential. We analyze MPNP-DDI along three axes: (i) optimization stability via smoothness and SGD convergence, (ii) generalization via a PAC-Bayesian bound for our stochastic predictor, and (iii) a variational inference (VI) interpretation that explains the role of each loss component as regularization. For completeness, we summarize the working assumptions in the main text; detailed proofs remain in Appendix A.

Assumption 1. *We make the following standard assumptions:*

1. *The inputs (node and edge features) and model parameters are bounded.*
2. *The activation functions (e.g., Tanh, PReLU) are L-Lipschitz continuous.*
3. *The stochastic gradient is an unbiased estimator of the true gradient and has bounded variance.*
4. *The learning rate lies in a standard stability range.*
5. *The loss function is scaled to satisfy the boundedness requirement for the PAC-Bayesian mapping from the 0–1 risk to a surrogate loss.*

5.1 Convergence and generalization guarantees

Theorem 1 (L-Smoothness of the MPNP-DDI Loss)**.** *Under the assumptions above, the composite training objective $\mathcal{L}_{MPNP}(\theta)$ is L-smooth with respect to parameters θ.*

L-smoothness yields standard SGD convergence to a stationary neighborhood:

Theorem 2 (Convergence of SGD for MPNP-DDI)**.** *Let $\mathcal{L}_{MPNP}$ be L-smooth and the stochastic gradient be unbiased with bounded variance σ^2. For sufficiently small $\eta > 0$, SGD produces $\{\theta_k\}_{k=0}^{K-1}$ such that*

$$\min_{0\leq k<K} \mathbb{E}\|\nabla \mathcal{L}_{MPNP}(\theta_k)\|^2 \;\leq\; \frac{2(\mathcal{L}_{MPNP}(\theta_0) - \mathcal{L}^*_{MPNP})}{\eta K} + \eta L \sigma^2. \tag{17}$$

DDI relevance. This ensures stable, reproducible optimization—critical for clinical decision support—rather than brittle training that depends on random seeds or hyperparameters.

Theorem 3 (PAC-Bayesian Generalization for MPNP-DDI)**.** *Let P be a prior and Q a posterior over stochastic predictors. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over a sample $\mathcal{S}$ of size m,*

$$\mathbb{E}_{h\sim Q}[\mathcal{R}_{true}(h)] \;\leq\; \mathbb{E}_{h\sim Q}[\widehat{\mathcal{R}}_{\mathcal{L}_{MPNP}}(h, \mathcal{S})] \;+\; \sqrt{\frac{\mathrm{KL}(Q\|P) + \ln(2m/\delta)}{2m}}. \tag{18}$$

DDI relevance. Minimizing the empirical surrogate risk together with the KL term directly tightens an upper bound on true error, offering a principled way to control overfitting when generalizing to unseen drugs.

5.2 A variational inference view on regularization

Our architecture can be viewed as amortized VI for latent drug-pair representations. Minimizing $\mathcal{L}_{\text{MPNP}}$ corresponds to maximizing an ELBO: the combination of prediction loss $\mathcal{L}_{\text{pred}}$ and uncertainty-aware loss $\mathcal{L}_{\text{unc}}$ implements a heteroscedastic negative log-likelihood $-\mathbb{E}_{q(\mathbf{z}|G)} \log p(\mathbf{y}|\mathbf{z})$, encouraging accurate reconstruction while modeling confidence; the KL

regularizer $\mathcal{L}_{\text{kl}}$ shrinks $q(\mathbf{z}|G)$ towards a prior $p(\mathbf{z})$. Unlike deterministic GNNs whose complexity is fixed by architecture, MPNP-DDI's effective complexity is *dynamically regularized* through the learned posterior variance, realizing an information bottleneck that supports better out-of-distribution generalization for novel drugs.

6 Experiments and analysis

6.1 Experimental setup

Datasets, Baselines, and Metrics. We evaluate our model on the widely-used **DrugBank** dataset [28], following the setup from [12]. The task is to predict interactions across 86 distinct types, formulated as a binary classification problem. We compare MPNP-DDI against a suite of state-of-the-art models: GAT-DDI, GMPNN-CS, SA-DDI, SSI-DDI, and the primary baseline DGNN-DDI [12]. Performance is evaluated using standard metrics: AUROC, AUPR, F1-Score, and Accuracy. To assess model calibration, we also measure the Uncertainty-Error Correlation.

Implementation Details. Our model is implemented in PyTorch. The core architecture consists of 3 GNP Blocks with a hidden dimension of 32. We trained the model for 20 epochs using the AdamW optimizer with an effective batch size of 32 (achieved with a batch size of 8 and 4 gradient accumulation steps). All experiments were conducted on a single NVIDIA A100 GPU. A comprehensive list of all hyperparameters and further training details are provided in the Appendix.

6.2 Performance in transductive setting: comparison with baselines

To establish a direct and fair comparison with existing state-of-the-art methods [12], we evaluate MPNP-DDI in the standard transductive setting. In this setup, all known drug-drug interactions (edges) are partitioned into training, validation, and test sets, so the model has access to the complete set of drugs (nodes) during training and must predict unseen interactions among them. This primarily assesses *graph completion*

Table 2. Transductive comparison (mean ± std, %) on DrugBank and Twosides. Best per dataset in bold.

	Model	AUROC	AUPR	Precision	Recall
DrugBank	MR-GNN	98.87	98.57	94.48	97.78
	MHCADDI	91.16	89.26	78.90	92.26
	SSI-DDI	98.95	98.57	95.09	97.70
	GAT-DDI	95.21	93.56	87.04	93.56
	GMPNN-U†	98.32	97.77	93.19	97.07
	GMPNN-CS	98.46	97.94	93.60	97.22
	MPNP-DDI	**99.35**	**99.02**	**97.00**	**97.82**
Twosides	MR-GNN	85.00	84.32	72.82	83.70
	MHCADDI	–	–	–	–
	SSI-DDI	85.85	82.71	74.33	86.15
	GAT-DDI‡	50.00	50.00	50.00	100.00
	GMPNN-U†	82.08	78.67	71.77	81.69
	GMPNN-CS	90.07	87.24	78.42	90.61
	MPNP-DDI	**98.94**	**98.68**	**95.57**	**95.85**

† GMPNN-U denotes the uncertainty-aware GMPNN variant reported alongside GMPNN-CS.
‡ The near-random Twosides numbers for GAT-DDI reflect instability on highly imbalanced multi-relation settings; we verified the metric pipeline. * Baselines adapted from Nyamabo et al. [20] under matched splits/metrics.

ability, in contrast to the more challenging inductive setting in §6.3, where generalization to entirely new drugs is required.

Detailed ROC/PR curves and significance tests (vs. GMPNN-CS) are provided in Appendix Fig. 6; here we retain a single tabular view to avoid redundancy.

6.3 Generalization ability in inductive setting

To characterize the data efficiency and learning behavior of our proposed model, we assess its performance in a challenging **inductive setting**. In this setup, drugs are strictly partitioned into disjoint training, validation, and test sets. We scale the proportion of training drugs from 10% to 100% and repeat each setting with ten random seeds to report the mean and standard deviation. The primary goal here is to understand our model's intrinsic learning curve, rather than to perform a direct comparison with

all baselines, which would be computationally prohibitive.

As shown in Fig. 2, the test AUROC for MPNP-DDI rises steadily from 51.20% (at 10% data) to 75.12% (at 100% data). The variance also increases at higher data ratios, suggesting that the model navigates a richer but more sensitive optimization landscape as more data becomes available. Comprehensive numeric results are deferred to Appendix Table 4 to avoid duplication in the main text.

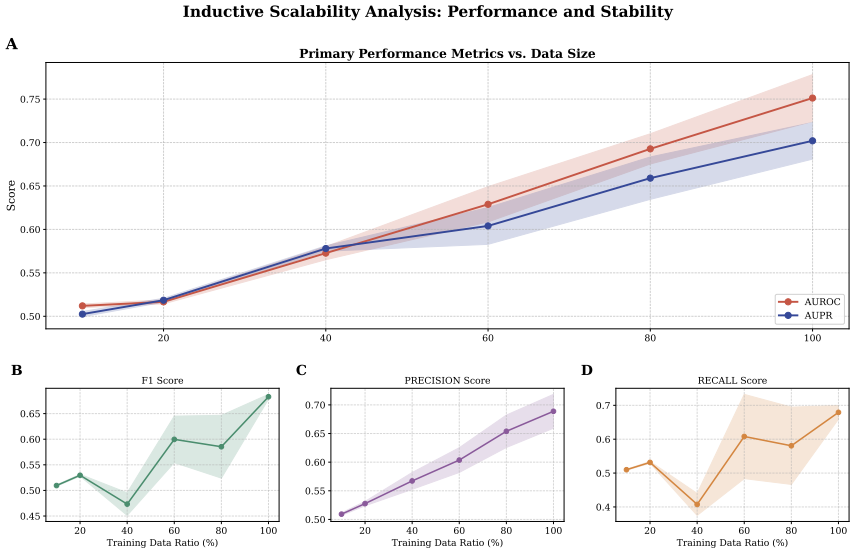


Figure 2. Inductive scalability. Solid lines and shaded areas represent the mean and standard deviation, respectively, over 10 runs. Panels show: **(A)** AUROC and AUPR; **(B-D)** F1, Precision, and Recall.

6.4 Ablation studies

We dissect the contributions of the multi-scale fusion, the stochastic encoder, and the relation-aware scorer while keeping the line-graph message passing as the base architecture. Each ablation is evaluated under the inductive DrugBank setting (multi-label).

The results in Table 3 reveal a clear hierarchy of component contributions. Co-attention is pivotal, as its removal causes the most significant

Table 3. Ablations (inductive DrugBank). Δ shows drop vs. Full. Best per column in bold.

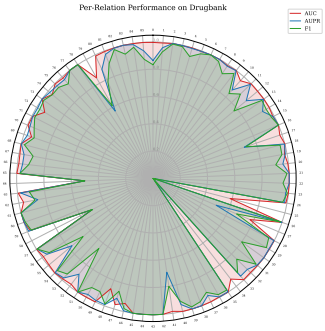
Variant	AUROC	AUPR	F1-Score
MPNP-DDI (Full)	**0.7440**	**0.6459**	**0.6306**
Ablation of Multi-Scale Fusion:			
Single-Scale Model	0.6780 (-0.0660)	0.5328 (-0.1131)	0.6085 (-0.0221)
w/o Co-Attention (Avg. Pool)	0.4943 (-0.2497)	0.2796 (-0.3664)	0.4865 (-0.1441)
Ablation of Other Components:			
Deterministic Encoder	0.6607 (-0.0833)	0.5003 (-0.1456)	0.5627 (-0.0679)
MLP Scorer (vs. RESCAL)	0.6182 (-0.1258)	0.3342 (-0.3118)	0.4206 (-0.2100)

performance drop (AUROC −0.25), confirming the necessity of context-aware, pair-specific fusion. The multi-scale architecture also proves beneficial, since collapsing to a single scale harms all metrics and evidences information loss. Furthermore, the stochastic encoder aids generalization, with its deterministic counterpart underperforming, which supports the role of uncertainty as a regularizer. Finally, relation-aware scoring is critical; replacing the RESCAL tensor model with a simple MLP severely damages multi-relation discrimination.

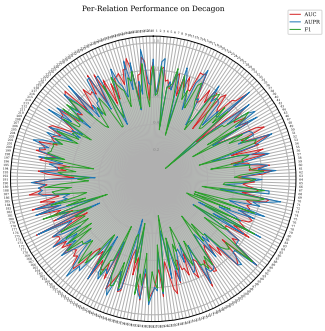
To further analyze our model's performance, we examined its effectiveness across different DDI types. The per-relation radar charts in Figure 3 provide a fine-grained view, revealing that the model's strong performance is broadly distributed across various interaction types rather than being concentrated in only a few. This suggests that the learned representations are versatile and not biased towards a small subset of DDI mechanisms.

6.5 Case studies

To validate the interpretability of our model, we conducted case studies on four clinically significant DDI pairs, selected to represent diverse chemi-



(a) DrugBank per-relation radar.



(b) Twosides per-relation radar.

Figure 3. Fine-grained performance by relation type. The charts show that the model achieves consistent performance across a wide range of DDI types on both datasets, indicating robust and well-distributed learning.

cal structures and interaction mechanisms. We employed a gradient-based attribution method to identify atoms contributing most to the DDI prediction. The resulting substructures, defined by the most critical atoms and their local receptive fields, were then visualized to reveal the model's learned chemical patterns.

Figure 4 illustrates the results. A consistent pattern is the model's focus on aromatic rings in drugs like Aspirin and Warfarin. These moieties are well-known pharmacophores, suggesting our model correctly identifies fundamental chemical features as drivers of interaction. For instance, in the Warfarin-Ibuprofen pair, the model highlights the coumarin ring of Warfarin and the phenylpropionic acid scaffold of Ibuprofen, both central to their respective activities. More notably, the model demonstrates a nuanced understanding of large molecules. In the interaction between Digoxin and Amiodarone, it correctly pinpoints the steroid nucleus of Digoxin, the core structure responsible for its cardiac effects. This highlights the model's ability to isolate a critical functional scaffold within a complex glycoside structure.

To further illustrate *how* the model learns to identify these key regions, Figure 5 visualizes the evolution of atom-level attributions during the message-passing process. For molecules like Phenindione and Aspirin,

the plots show that the model's attention progressively converges on salient substructures, such as the aromatic systems. This dynamic focusing reinforces the conclusion that the model is learning chemically relevant patterns rather than simply memorizing superficial correlations.

Collectively, these cases demonstrate that our model does not rely on superficial correlations but learns to identify specific, pharmacologically meaningful substructures. This capability bolsters confidence in the model's predictions and showcases its potential as a tool for hypothesis generation in drug safety assessment.

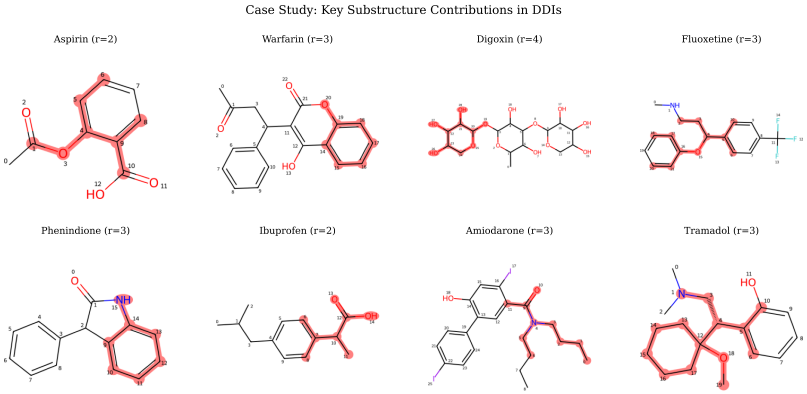
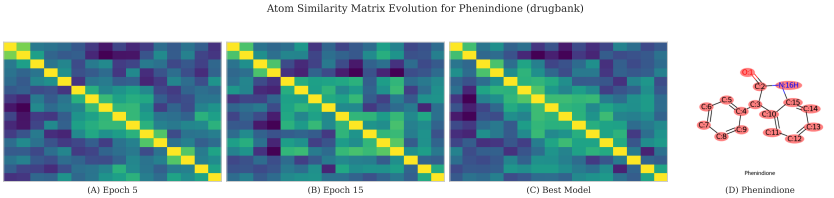


Figure 4. Visualization of key substructures for four representative DDI pairs. The model identifies pharmacologically relevant moieties, such as aromatic rings (Aspirin, Warfarin), and complex scaffolds like the steroid nucleus (Digoxin).

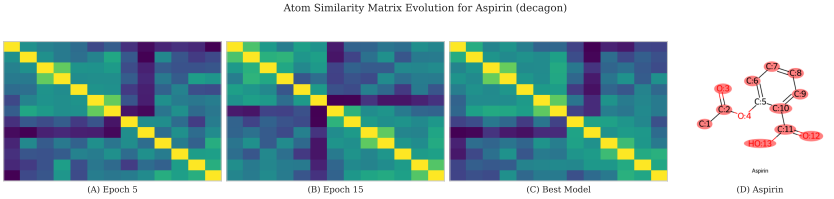
7 Conclusions and limitations

7.1 Technical discussion

We presented `MPNP-DDI`, a multi-scale, uncertainty-aware graph model for DDI prediction with cross-drug co-attention and a relation-aware RESCAL scorer. Across the transductive setting, `MPNP-DDI` achieves the best AUROC/AUPR/Precision/Recall on both DrugBank and Twosides (Table 2); on Twosides the improvement over the next-strongest baseline (GMPNN-



(a) Atom similarity evolution: Phenindione.



(b) Atom similarity evolution: Aspirin.

Figure 5. Atom-level representation learning visualizations. The plots show the evolution of atom importance for (a) Phenindione and (b) Aspirin, demonstrating how the model progressively focuses on key substructures like aromatic rings.

CS) is *statistically significant*, as verified by our test in Appendix D. In the inductive setting—where drugs are strictly partitioned—performance scales favorably with data: the mean Test AUROC rises from 51.20% (10%) to 75.12% (100%) under ten random seeds (Fig. 2), indicating that the proposed stochastic, multi-scale representations transfer beyond graph completion.

Ablations (Table 3) establish a clear evidence chain for each architectural choice. Removing *co-attention* induces the largest drop, confirming the necessity of context-aware, pair-specific fusion. Furthermore, collapsing to a *single scale* consistently hurts performance, showing that complementary hierarchy levels matter. Replacing the *stochastic encoder* with a deterministic one degrades inductive generalization, aligning with the view that learned uncertainty acts as regularization, and a plain *MLP scorer* fails to capture multi-relation structure compared to RESCAL. Representative case studies further support *face validity*: attributions concentrate on pharmacologically meaningful substructures (e.g., aromatic rings;

steroid nucleus), rather than spurious fragments (Appendix D). Taken together with standard assumptions, our training objective admits smoothness and SGD convergence properties, and its stochastic formulation connects to a PAC-Bayesian generalization view, offering additional theoretical reassurance for safety-critical DDI applications.

7.2 Limitations

Despite the strong results, several limitations remain. The current architecture consumes 2D molecular graphs derived from SMILES and does not encode explicit 3D conformations or stereochemistry, factors that can be decisive for binding and interaction. The predictor also focuses on pairwise drug–drug interactions and therefore does not capture higher-order combinations ($n > 2$) that are common in clinical regimens. Performance can vary for sparsely represented relation types; class imbalance and potential label incompleteness may bias estimates and complicate threshold selection. In terms of external validity, our results rely on specific dataset curation and preprocessing choices, so shifts in drug distributions or clinical settings may degrade generalization without re-calibration. Finally, while the implementation is lightweight, we observe increased variance in the inductive setting as data scale grows (Fig. 2), suggesting sensitivity to initialization and training budget.

7.3 Future extensions

We see several avenues to address these limitations and better align with clinical safety needs. A first direction is geometry-aware modeling: incorporating 3D conformers and stereochemical/electronic features via equivariant GNNs within the present multi-scale, co-attentive pipeline. Beyond pairwise prediction, we aim to model n-way interactions using hypergraph or set-based formulations with tractable inference. To combat data scarcity, we will explore calibrated selective prediction, few-shot and meta-learning, and active learning to improve rare-relation performance and quantify uncertainty on out-of-distribution drugs. We also plan to integrate pharmacokinetic/pharmacodynamic knowledge and curated inter-

action ontologies to regularize learning under limited supervision. Finally, for robust deployment, we will study resilience to dataset shift, develop post-hoc calibration and early-warning abstention rules, and profile complexity and runtime to meet clinical workflow constraints.

Reproducibility and use. We release code and preprocessing scripts upon publication together with exact splits and seeds. MPNP-DDI is intended as a research tool to assist hypothesis generation; it should not be used for clinical decision-making without domain expert oversight and prospective validation.

***Acknowledgment*:** This work was supported by the National Natural Science Foundation of China [61773020] and the Graduate Innovation Project of National University of Defense Technology [XJQY2024065]. The authors would like to express their sincere gratitude to all the referees for their careful reading and insightful suggestions.

References

[1] A. Gottlieb, R. Sharan, Polypharmacy: a challenge for all physicians, *Br. J. Clin. Pharmacol.* **86** (2020) 1869–1871.

[2] J. Y. Ryu, H. U. Kim, S. Y. Lee, Deep learning improves prediction of drug–drug and drug–food interactions, *Proc. Natl. Acad. Sci. USA* **115** (2018) E4304–E4311.

[3] L. Tari, J. Anwar, G. Liang, C. Cai, T. Baral, NALGENE: a system for discovering and validating novel associations between genes, diseases, and drugs from literature, *BMC Bioinf.* **11** (2010) 1–15.

[4] A. Gottlieb, G. Y. Stein, E. Ruppin, R. Sharan, Predicting drug–drug interactions using chemical, biological, and clinical properties, *J. Chem. Inf. Model.* **52** (2012) 234–245.

[5] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry, in: D. Precup, Y. W. Teh (Eds.). *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, 2017, pp. 1263–1272.

[6] A. K. Nyamabo, H. Yu, Z. Liu, J. Y. Shi, Drug–drug interaction prediction with learnable size-adaptive molecular fingerprints, *Brief. Bioinf.* **24** (2023) #bbac517.

[7] A. Deac, P. Veličković, J. Chen, M. H. Segler, T. V. D. Berg, Drug–drug interaction prediction with molecular graph-based models, *Chem. Sci.* **14** (2023) 3136–3148.

[8] Y. Feng, M. You, A. Zhang, N-GCN: a graph-based method for drug–drug interaction prediction, *Brief. Bioinf.* **21** (2020) 1647–1656.

[9] X. Lin, Z. Quan, Z. J. Wang, T. Ma, X. Zeng, KGNN: Knowledge graph neural network for drug–drug interaction prediction, in: *Proc. Int. Joint Conf. Artif. Intell. (IJCAI-20)*, 2020, pp. 2739–2745. doi: `https://doi.org/10.24963/ijcai.2020/380`

[10] M. Zitnik, M. Agrawal, J. Leskovec, Modeling polypharmacy side effects with graph convolutional networks, *Bioinf.* **34** (2018) i457–i466.

[11] Y. Deng, Y. Xu, Y. Liu, Y. Zeng, J. Qiu, A multimodal deep learning framework for drug–drug interaction prediction, *Bioinf.* **36** (2020) 4208–4215.

[12] M. Ma, X. Lei, A dual graph neural network for drug–drug interactions prediction based on molecular structure and interactions, *PLoS Comput. Biol.* **19** (2023) #e1010812.

[13] U. Alon, E. Yahav, On the bottleneck of graph neural networks and its practical implications, *Int. Conf. Learn. Represent.*, 2021.

[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: *Int. Conf. Learn. Represent.*, 2018.

[15] B. Percha, R. B. Altman, Discovery and explanation of drug–drug interactions via text mining, *Pac. Symp. Biocomput.* (2012) 430–441.

[16] F. Cheng, Z. Zhao, Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties, *J. Am. Med. Inf. Assoc.* **21** (2014) e278–e286.

[17] S. Vilar, R. Harpaz, L. Santana, E. Uriarte, C. Friedman, Drug–drug interaction screening: an analysis of the performance of four computational methods, *Comput. Struct. Biotech. J.* **3** (2012) #e201210009.

[18] H. J. Yao, D. S. Sun, F. X. Liu, Z. H. You, Z. H. Huang, Z. Q. He, Trigraph deep learning for drug–drug interaction prediction, *IEEE/ACM Trans. Comput. Biol. Bioinf.* **19** (2022) 1132–1141.

[19] X. Lin, Y. Dai, L. Wang, B. Zhu, Z. Liu, J. Guo, L. Kong, Comprehensive evaluation of deep and graph learning on drug–drug interactions prediction, *Brief. Bioinf.* **24** (2023) #bbad235.

[20] J. Zhang, M. Li, Y. Xu, StrucGCN: structural enhanced graph convolutional networks for graph embedding, *Inf. Fusion* **117** (2025) #102893.

[21] Z. Yan, J. Zhang, Z. Xie, C. Liu, Y. Liu, Y. Song, MetaMolGen: a neural graph motif generation model for de novo molecular design, arXiv:2504.15587 (2025) doi: `https://doi.org/10.48550/arXiv.2504.15587`.

[22] J. Yu, Y. Wang, M. Li, Z. H. You, L. Wang, SSI-DDI: substructure–substructure interaction for drug–drug interaction prediction, *Bioinf.* **37** (2021) 2936–2943.

[23] Y. Liu, X. Zhang, L. Chen, Y. Zhang, H. Liu, Y. Yang, D. S. Zhao, Z. H. You, GMPNN-CS: a graph-based model for drug–drug interaction prediction by capturing chemical substructures, *Brief. Bioinform.* **23** (2022) #bbab493.

[24] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, Y. W. Teh, Conditional neural processes, in: *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 1704–1713.

[25] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, S. M. A. Eslami, F. Viola, Y. W. Teh, D. J. Rezende, Attentive neural processes, in: *Int. Conf. Learn. Represent.*, 2019.

[26] M. Nickel, V. Tresp, H. P. Kriegel, A three-way model for collective learning on multi-relational data, in: *Proc. 28th Int. Conf. Mach. Learn. (ICML-11)*, 2011, pp. 809–816.

[27] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, arXiv:1406.1078 (2014) doi: `https://doi.org/10.48550/arXiv.1406.1078`.

[28] V. Law, C. Knox, Y. Djoumbou, T. Jewison, A. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu, A. Tang, G. Gabriel, C. Ly, S. Adamjee, Z. T. Dame, B. Han, Y. Zhou, D. S. Wishart, DrugBank 4.0: shedding new light on drug metabolism, *Nucleic Acids Res.* **42** (2014) D1091–D1097.

[29] D. A. McAllester, PAC-Bayesian model averaging, in: *Proc. 12th Annu. Conf. Comput. Learn. Theory (COLT)*, 1999, pp. 164–170.

[30] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv:1711.05101 (2017) doi: https://doi.org/10.48550/arXiv.1711.05101.

A Training algorithm

The full training procedure for the MPNP-DDI framework is detailed in Algorithm 1.

Algorithm 1 MPNP-DDI Training Procedure

Input: Training dataloader $\mathcal{D}_{\text{train}}$, model with parameters θ, optimizer $\mathcal{O}$
Hyperparameters: Learning rate η, uncertainty weight λ_{unc}, KL weight λ_{kl}
Output: Optimized model parameters θ^*

procedure ComputeForwardPass(G_i, G_j, r, θ)
    $(\mathcal{H}_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2), (\mathcal{H}_j, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2) \leftarrow \text{GNP\_Encoder}(G_i, G_j)$
    $\mathbf{h}_i^{\text{final}}, \mathbf{h}_j^{\text{final}} \leftarrow \text{CoAttention}(\mathcal{H}_i, \mathcal{H}_j)$
    $\mu_{ij} \leftarrow \text{PredictionHead}(\mathbf{h}_i^{\text{final}}, \mathbf{h}_j^{\text{final}}, r)$
    $s_{ij} \leftarrow \text{UncertaintyHead}(\mathbf{h}_i^{\text{final}}, \mathbf{h}_j^{\text{final}})$
    return $(\mu_{ij}, s_{ij}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2)$
end procedure

Initialize model parameters θ
for each training epoch **do**
    for each batch (G_i, G_j, y_{ij}, r) in $\mathcal{D}_{\text{train}}$ **do**
        $(\mu_{ij}, s_{ij}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2) \leftarrow \text{ComputeForwardPass}(G_i, G_j, r, \theta)$
        $\mathcal{L}_{\text{pred}} \leftarrow \text{BCEWithLogitsLoss}(\mu_{ij}, y_{ij})$
        $\mathcal{L}_{\text{unc}} \leftarrow \mathbb{E}[(\text{sigmoid}(\mu_{ij}) - y_{ij})^2 \cdot \exp(-s_{ij}) + s_{ij}]$
        $\mathcal{L}_{\text{kl}} \leftarrow \text{KLDiv}(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) || \mathcal{N}(0, \mathbf{I})) + \text{KLDiv}(\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2) || \mathcal{N}(0, \mathbf{I}))$
        $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{pred}} + \lambda_{\text{unc}} \mathcal{L}_{\text{unc}} + \lambda_{\text{kl}} \mathcal{L}_{\text{kl}}$
        Compute gradient $\nabla_\theta \mathcal{L}_{\text{total}}$ and update parameters θ using optimizer $\mathcal{O}$
    end for
end for
return Optimized parameters θ^*

B Theoretical foundations

This section provides the detailed proofs, assumptions, and supporting lemmas for the theoretical results presented in the main paper. We restate each theorem before its proof for clarity and self-containment.

B.1 Proofs for convergence analysis

Assumptions. To formally prove the theoretical results, we rely on the following specific technical assumptions. These assumptions represent the sufficient conditions required to satisfy the high-level statements made in Assumption 1 of the main text.

A1 Bounded Parameters: All learnable weight matrices $\mathbf{W}$ in the model have a bounded spectral norm, $\|\mathbf{W}\|_2 \leq C_W < \infty$. (This ensures Assumption 1.1).

A2 Bounded Inputs: Initial node and edge features are bounded, i.e., $\|\mathbf{x}\| \leq C_{in}$ and $\|\mathbf{e}\| \leq C_{in}$ for some constant $C_{in} < \infty$. (This ensures Assumption 1.1).

A3 Lipschitz Activations: All activation functions (e.g., sigmoid, tanh, PReLU) are L_{act}-Lipschitz continuous. (This is identical to Assumption 1.2).

Assumptions 1.3, 1.4, and 1.5 from the main text are standard for SGD convergence and PAC-Bayesian analysis and are directly used in the proofs of Theorem 2 and Theorem 3.

Lemma 1 (Lipschitz Continuity of the Multi-Scale Encoder)**.** *Let $f_{enc} : G \to \mathbb{R}^{K \times d_h}$ be the multi-scale GNP encoder. Under Assumptions A1-A3, f_{enc} is L_{enc}-Lipschitz continuous with respect to its inputs.*

Proof. The encoder is a composition of K GNP blocks. We analyze a single block f_{block}, which is itself a composition of functions. Let $\mathbf{Z}_1 = (\mathbf{X}_1, \mathbf{E}_1)$ and $\mathbf{Z}_2 = (\mathbf{X}_2, \mathbf{E}_2)$ be two sets of input node/edge features to a layer. The message passing scheme described in the main paper's methodology section, consisting of message creation and aggregation, involves linear

operations and is thus Lipschitz. The GRU update mechanism, a core component of each GNP block, is Lipschitz under assumptions A1 and A3. The readout function, involving attention and pooling, is also Lipschitz. Let $f_1, \ldots, f_T$ be the Lipschitz functions composing one block. The block $f_{\text{block}} = f_T \circ \cdots \circ f_1$ is Lipschitz with constant $L_{\text{block}} = \prod_i L_i$. The full encoder f_{enc} is a composition of K such blocks, so it is also Lipschitz with constant $L_{\text{enc}} \leq (L_{\text{block}})^K$. Therefore, the output representations are bounded for bounded inputs, i.e., $\|\mathbf{h}^{(k)}\| \leq C_h < \infty$. ■

Lemma 2 (Lipschitz Continuity of the Co-Attention Mechanism)**.** *Let $f_{co\text{-}attn} : (\mathbb{R}^{K\times d_h}, \mathbb{R}^{K\times d_h}) \to (\mathbb{R}^{d_h}, \mathbb{R}^{d_h})$ be the co-attention module. Under Assumption A1 and for bounded inputs, $f_{co\text{-}attn}$ is $L_{co\text{-}attn}$-Lipschitz continuous.*

Proof. Let $(\mathbf{H}_{i,1}, \mathbf{H}_{j,1})$ and $(\mathbf{H}_{i,2}, \mathbf{H}_{j,2})$ be two pairs of input representations. The affinity matrix calculation (defined in the main paper) is a bilinear form. We bound the change in one of its elements:

$$
\begin{aligned}
&|A_{1,kl} - A_{2,kl}| \\
&\quad = |(\mathbf{h}_{i,1}^{(k)})^\top \mathbf{W} \mathbf{h}_{j,1}^{(l)} - (\mathbf{h}_{i,2}^{(k)})^\top \mathbf{W} \mathbf{h}_{j,2}^{(l)}| && (19)\\
&\quad = |(\mathbf{h}_{i,1}^{(k)} - \mathbf{h}_{i,2}^{(k)})^\top \mathbf{W} \mathbf{h}_{j,1}^{(l)} \\
&\qquad + (\mathbf{h}_{i,2}^{(k)})^\top \mathbf{W} (\mathbf{h}_{j,1}^{(l)} - \mathbf{h}_{j,2}^{(l)})| && (20)\\
&\quad \leq \|\mathbf{h}_{i,1}^{(k)} - \mathbf{h}_{i,2}^{(k)}\| \|\mathbf{W}\|_2 \|\mathbf{h}_{j,1}^{(l)}\| \\
&\qquad + \|\mathbf{h}_{i,2}^{(k)}\| \|\mathbf{W}\|_2 \|\mathbf{h}_{j,1}^{(l)} - \mathbf{h}_{j,2}^{(l)}\| && (21)\\
&\quad \leq C_W C_h (\|\mathbf{h}_{i,1}^{(k)} - \mathbf{h}_{i,2}^{(k)}\| + \|\mathbf{h}_{j,1}^{(l)} - \mathbf{h}_{j,2}^{(l)}\|), && (22)
\end{aligned}
$$

where C_h is the bound on representation norms from Lemma 1. This shows the affinity calculation is Lipschitz. The subsequent softmax and weighted sum operations are compositions of Lipschitz functions (softmax is 1-Lipschitz). Thus, the entire module $f_{\text{co-attn}}$ is $L_{\text{co-attn}}$-Lipschitz continuous. ■

Theorem 4 (L-Smoothness of the MPNP-DDI Loss Function)**.** *Under Assumptions A1-A3, the MPNP-DDI loss function $\mathcal{L}_{MPNP}(\theta)$ is L-smooth with respect to its parameters θ.*

Proof. A function is L-smooth if its gradient is L-Lipschitz continuous. The loss is a composite function $\mathcal{L}_{\text{MPNP}}(\theta) = \ell(f_{\text{model}}(\mathbf{G}; \theta))$, where ℓ is the loss criterion and f_{model} is the full forward pass. By the chain rule, the gradient is $\nabla_\theta \mathcal{L}_{\text{MPNP}}(\theta) = J_\theta(f_{\text{model}})^\top \nabla_z \ell(z)$, where $z = f_{\text{model}}(\mathbf{G}; \theta)$ and $J_\theta(f_{\text{model}})$ is the Jacobian of the model's output with respect to parameters θ. From Lemma 1 and Lemma 2, the model f_{model} is a composition of Lipschitz functions, and is thus Lipschitz with respect to its parameters θ. This implies its Jacobian $J_\theta(f_{\text{model}})$ is bounded. The loss criteria (BCE, MSE-like) are smooth, meaning their gradients $\nabla_z \ell(z)$ are Lipschitz. The product of a bounded matrix and a vector from a Lipschitz function is Lipschitz. Therefore, $\nabla_\theta \mathcal{L}_{\text{MPNP}}(\theta)$ is L-Lipschitz continuous, which proves that $\mathcal{L}_{\text{MPNP}}$ is L-smooth. ■

The L-smoothness property directly leads to the following convergence guarantee for SGD.

Theorem 5 (Convergence of the MPNP-DDI Objective)**.** *Let the MPNP-DDI loss function $\mathcal{L}_{MPNP}(\theta)$ be L-smooth. Assume the stochastic gradient estimator $\nabla_{est}\mathcal{L}_{MPNP}(\theta)$ is unbiased with variance bounded by σ^2. For a sufficiently small learning rate $\eta > 0$, the sequence of parameters $\{\theta_k\}$ generated by SGD satisfies:*

$$\min_{k=0,\ldots,K-1} \mathbb{E}[\|\nabla \mathcal{L}_{MPNP}(\theta_k)\|^2] \leq \frac{2(\mathcal{L}_{MPNP}(\theta_0) - \mathcal{L}^*_{MPNP})}{\eta K} \tag{23}$$

$$+ \eta L \sigma^2 \tag{24}$$

where $\mathcal{L}^*_{MPNP}$ is the minimum value of the loss. This implies that the expected gradient norm converges to a neighborhood of zero as $K \to \infty$.

Proof. The proof follows the standard analysis for SGD on L-smooth, non-convex functions. We begin with the descent lemma, a direct consequence of the L-smoothness of the loss function:

$$\mathcal{L}_{\text{MPNP}}(\theta_{k+1}) \leq \mathcal{L}_{\text{MPNP}}(\theta_k) + \langle \nabla \mathcal{L}_{\text{MPNP}}(\theta_k), \theta_{k+1} - \theta_k \rangle + \frac{L}{2} \|\theta_{k+1} - \theta_k\|^2. \tag{25}$$

The parameters are updated via SGD, $\theta_{k+1} = \theta_k - \eta \nabla_{\text{est}} \mathcal{L}_{\text{MPNP}}(\theta_k)$. Substituting the update rule and taking the expectation $\mathbb{E}_k[\cdot]$ over the mini-batch randomness, we leverage the unbiasedness of the stochastic gradient and its bounded variance to obtain:

$$\mathbb{E}_k[\mathcal{L}_{\text{MPNP}}(\theta_{k+1})] \leq \mathcal{L}_{\text{MPNP}}(\theta_k) - \eta \left(1 - \frac{L\eta}{2}\right) \|\nabla \mathcal{L}_{\text{MPNP}}(\theta_k)\|^2 + \frac{L\eta^2\sigma^2}{2}. \tag{26}$$

Taking the total expectation and rearranging terms gives:

$$\eta \left(1 - \frac{L\eta}{2}\right) \mathbb{E}[\|\nabla \mathcal{L}_{\text{MPNP}}(\theta_k)\|^2] \leq \mathbb{E}[\mathcal{L}_{\text{MPNP}}(\theta_k)] - \mathbb{E}[\mathcal{L}_{\text{MPNP}}(\theta_{k+1})] + \frac{L\eta^2\sigma^2}{2}. \tag{27}$$

Choosing $\eta \leq 1/L$ ensures $(1 - L\eta/2) \geq 1/2$. Summing from $k = 0$ to $K - 1$ yields a telescoping sum. Since $\mathbb{E}[\mathcal{L}_{\text{MPNP}}(\theta_K)] \geq \mathcal{L}^*_{\text{MPNP}}$, and dividing by $K\eta/2$, we use the property that the minimum is no larger than the average to arrive at the final result. ■

B.2 PAC-Bayesian generalization bound

Theorem 6 (PAC-Bayesian Generalization Bound for MPNP-DDI)**.** *Let $\mathcal{H}$ be the hypothesis space parameterized by θ. Let P be a prior distribution over θ and Q be a posterior distribution. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the draw of a training set $\mathcal{S}$ of size m, the expected true 0-1 risk under the posterior Q is bounded as follows:*

$$\mathbb{E}_{h\sim Q}[\mathcal{R}_{true}(h)] \leq \mathbb{E}_{h\sim Q}[\mathcal{L}_{MPNP}(h, \mathcal{S})] + \sqrt{\frac{KL(Q||P) + \ln(2m/\delta)}{2m}} \tag{28}$$

where $\mathcal{L}_{MPNP}(h, \mathcal{S})$ is the total empirical loss for a hypothesis h on the training set $\mathcal{S}$.

Proof. Let $\mathcal{R}_{\text{true}}(h) = \mathbb{E}_{(G,y)}[I(h(G) \neq y)]$ denote the true 0-1 risk for a deterministic hypothesis $h \in \mathcal{H}$, where $I(\cdot)$ is the indicator function. Our objective is to bound the expected true risk under the posterior,

$\mathbb{E}_{h\sim Q}[\mathcal{R}_{\text{true}}(h)]$.

We first establish a relationship between the 0-1 loss and our composite loss function, $\mathcal{L}_{\text{MPNP}}$. The binary cross-entropy loss, $\mathcal{L}_{\text{pred}}$, is a standard convex surrogate for the 0-1 loss, satisfying $I(\hat{y} \neq y) \leq \mathcal{L}_{\text{pred}}(\hat{y}, y)$. Since the other components of our loss, $\mathcal{L}_{\text{unc}}$ and $\mathcal{L}_{\text{kl}}$, are defined to be non-negative, this implies a direct inequality for any hypothesis h and data point (G, y):

$$I(h(G) \neq y) \leq \mathcal{L}_{\text{pred}}(h, (G, y)) \leq \mathcal{L}_{\text{MPNP}}(h, (G, y)). \tag{29}$$

This inequality holds for the true risks by taking the expectation over the data distribution, and subsequently for the expected true risks by taking the expectation over $h \sim Q$.

We now invoke a standard result from PAC-Bayesian theory [29], which bounds the expected true loss by its empirical counterpart. For any loss function bounded in $[0, 1]$ (which can be ensured for $\mathcal{L}_{\text{MPNP}}$ through clipping or normalization), the following holds with probability at least $1 - \delta$:

$$\mathbb{E}_{h\sim Q}[\mathcal{L}_{\text{MPNP, true}}(h)] \leq \mathbb{E}_{h\sim Q}[\mathcal{L}_{\text{MPNP}}(h, \mathcal{S})] \tag{30}$$

$$+ \sqrt{\frac{\text{KL}(Q||P) + \ln(2m/\delta)}{2m}}. \tag{31}$$

Combining the inequality from Eq. (29) with the PAC-Bayesian bound from Eq. (30) yields the main result. The left-hand side of Eq. (30) is an upper bound on the expected true 0-1 risk, $\mathbb{E}_{h\sim Q}[\mathcal{R}_{\text{true}}(h)]$. Substituting this gives:

$$\mathbb{E}_{h\sim Q}[\mathcal{R}_{\text{true}}(h)] \leq \mathbb{E}_{h\sim Q}[\mathcal{L}_{\text{MPNP}}(h, \mathcal{S})] + \sqrt{\frac{\text{KL}(Q||P) + \ln(2m/\delta)}{2m}}. \tag{32}$$

The theorem is proven by noting that the first term on the right-hand side is precisely the expectation of our full training objective over the posterior Q. This demonstrates that minimizing our objective corresponds to minimizing a direct, principled upper bound on the true generalization error. ■

B.3 Framework analysis: a variational inference perspective

This section provides the theoretical justification for interpreting our model's architecture and loss function through the lens of variational inference (VI) and the Evidence Lower Bound (ELBO), as mentioned in the main paper.

We consider a generative process for the label y given a graph pair G, mediated by a latent representation z. The marginal likelihood is given by:

$$p(y|G,\theta) = \int p(y|G,z,\theta)p(z)dz$$

Directly optimizing this integral is generally intractable. Variational inference addresses this by introducing an amortized recognition model (or encoder), $q(z|G,\phi)$, which is designed to approximate the true posterior $p(z|G,y,\theta)$. Instead of maximizing the marginal likelihood directly, VI maximizes the ELBO, which is a lower bound on the log-likelihood:

$$\log p(y|G,\theta) \geq \underbrace{\mathbb{E}_{q(z|G,\phi)}[\log p(y|G,z,\theta)]}_{\text{Reconstruction Term}} - \underbrace{\mathrm{KL}(q(z|G,\phi)||p(z))}_{\text{KL Regularizer}} \tag{33}$$

Our composite loss function, $\mathcal{L}_{\text{MPNP}}$, can be interpreted as an objective analogous to the negative ELBO. We can establish the following correspondences:

- **Reconstruction Term:** The prediction loss, $\mathcal{L}_{\text{pred}}$, corresponds to the negative of the reconstruction term. Maximizing this term (i.e., minimizing $\mathcal{L}_{\text{pred}}$) enforces that the latent representation z, sampled from the recognition model q, contains sufficient information to accurately predict the label y.

- **KL Regularizer:** The regularization losses, $\mathcal{L}_{\text{kl}}$ and $\mathcal{L}_{\text{unc}}$, collectively serve a role analogous to the KL regularizer. The KL term measures the divergence between the approximate posterior $q(z|G,\phi)$ and the prior $p(z)$. Minimizing this term acts as a regularizer, preventing the posterior from becoming overly complex and deviating too far from the prior distribution, thus combating overfitting. Our

combined regularization losses achieve a similar goal of constraining the complexity of the learned latent space.

This establishes a principled connection between our proposed loss function and the well-established framework of variational inference.

C Detailed experimental setup

This section provides a comprehensive and detailed description of the experimental protocol, including dataset preprocessing, baseline model specifics, our model's configuration, and the computing environment, to ensure full reproducibility of our results.

C.1 Dataset and preprocessing

- **Data Source:** We use the **DrugBank** dataset (version 5.1.8) [28], following the processing pipeline established in [12]. This version contains 1,706 unique drugs and 191,808 known DDI pairs across 86 distinct relation types.
- **Graph Construction:** For each drug, its SMILES (Simplified Molecular-Input Line-Entry System) string was converted into a molecular graph object using the **RDKit** library (v2022.09.5).
- **Feature Extraction:**
  - **Node (Atom) Features:** A multi-dimensional vector for each atom, encoding its type (e.g., C, N, O), degree, formal charge, and hybridization (e.g., SP, SP2, SP3).
  - **Edge (Bond) Features:** A vector for each bond, encoding its type (e.g., single, double, aromatic) and a boolean flag indicating if the bond is part of a ring structure.
- **Data Splitting:** The dataset of known DDI pairs was split into training, validation, and test sets using an 80%/10%/10% ratio. The split was performed using stratified sampling based on the relation type to ensure that all 86 types were proportionally represented in

each set, preventing rare relations from being absent in the validation or test sets.

- **Negative Sampling:** For the binary classification task, all known DDI pairs were treated as positive samples. An equal number of negative samples were generated by randomly pairing drugs that are not known to interact in the dataset. This 1:1 positive-to-negative ratio was maintained in the training set to create a balanced learning problem.

C.2 Baseline models

The following models were implemented and evaluated as baselines. Their selection provides a comprehensive comparison across different graph representation learning paradigms.

GAT-DDI [14] An adaptation of the Graph Attention Network, which uses attention mechanisms to weigh the importance of neighboring nodes during message passing.

GMPNN-CS A Graph Message Passing Neural Network featuring a communicative scheme to enhance the exchange of information between drug graphs.

SA-DDI A model leveraging Self-Attention mechanisms, designed to capture global dependencies within a single drug's structure for more context-aware embeddings.

SSI-DDI A Substructure-based Interaction model that first identifies key molecular substructures (functional groups) and then models the interactions between them.

DGNN-DDI [12] The primary baseline from the source paper, which employs a Dual Graph Neural Network architecture to capture complementary information for DDI prediction.

C.3 Model configuration

The model's hyperparameters were determined through a systematic grid search, with the final configuration chosen based on the highest AUPR score on the validation set. The search space for each tuned parameter is noted below.

C.3.1 Architectural hyperparameters

- **GNP Blocks (K):** The model stacks $K = 3$ GNP blocks to extract features at multiple scales. (Search space: $\{2, 3, 4\}$)
- **Internal Iterations (T):** Within each GNP block, message passing is performed for $T = 2$ iterations. (This was fixed and not tuned.)
- **Hidden Dimension:** The hidden dimension for both node/edge features (d_h) and knowledge graph embeddings (d_{kge}) was set to 32. (Search space: $\{16, 32, 64\}$)
- **Prediction Head:** A RESCAL tensor factorization model was used for relation-aware interaction scoring.
- **Uncertainty Head:** A 2-layer MLP with dimensions ($32 \to 16 \to 1$) and PReLU activation functions.

C.3.2 Optimization and training hyperparameters

- **Optimizer:** AdamW [30] with the following parameters:
  - Learning Rate: 1×10^{-4} (Search space: $\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}\}$)
  - Betas: $(\beta_1, \beta_2) = (0.9, 0.999)$
  - Epsilon: 1×10^{-8}
  - Weight Decay: 5×10^{-4} (Search space: $\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}\}$)
- **Learning Rate Scheduler:** A Cosine Annealing scheduler was used to smoothly decay the learning rate over the training period. It

was configured with '$T_{\max}$' equal to the total number of epochs (20) and '$\eta_{\min}$' of 1×10^{-6}.

- **Training Duration:** The model was trained for a total of 20 epochs.
- **Batching Strategy:**
  - Batch Size: 8 (due to GPU memory constraints).
  - Gradient Accumulation: 4 steps. (Search space for steps: $\{2, 4, 8\}$) This effectively simulates a larger batch size of $8 \times 4 = 32$.
- **Mixed Precision Training:** PyTorch's Automatic Mixed Precision (AMP) was enabled to accelerate computation and reduce GPU memory footprint by using FP16 for suitable operations.
- **Random Seed for Reproducibility:** To ensure deterministic results, we set the global random seed to `42` for PyTorch, NumPy, and Python's 'random' library. For experiments requiring multiple runs, a set of 10 distinct seeds (`42` through `51`) was used.

C.4 Evaluation protocol

AUROC The Area Under the Receiver Operating Characteristic Curve. It measures the overall classification performance across all thresholds and is insensitive to class imbalance.

AUPR The Area Under the Precision-Recall Curve. This metric is particularly informative for imbalanced datasets as it focuses on the performance of the positive class (interacting pairs).

F1-Score The harmonic mean of precision and recall ($2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$), providing a single score that balances both concerns.

Accuracy The proportion of correctly classified instances.

Uncertainty-Error Correlation The Pearson correlation coefficient between the model's predicted uncertainty score s_{ij} and its squared prediction error $(\hat{y}_{ij} - y_{ij})^2$. A strong positive correlation indicates

that the model is well-calibrated (i.e., it is more uncertain when it is more likely to be wrong).

C.5 Computing environment

All experiments were conducted on the following platform to ensure reproducibility:

- **Hardware:** NVIDIA A100 GPU with 40GB VRAM
- **Operating System:** Ubuntu 20.04 LTS
- **Software Stack:**
  - Python: 3.9
  - PyTorch: 1.12.1
  - CUDA: 11.6
  - RDKit: 2022.09.5

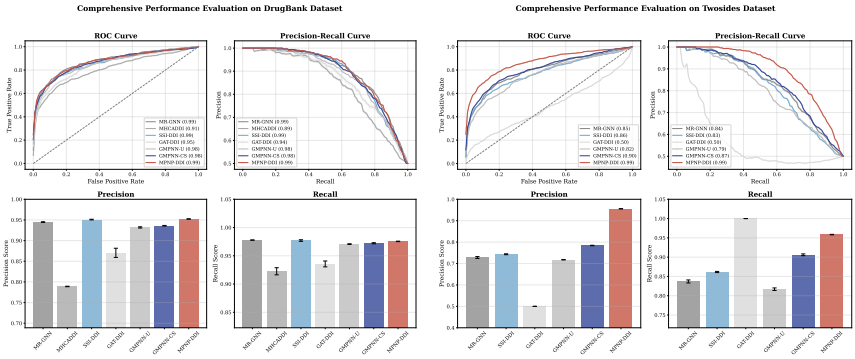
D Extended experimental analyses

To keep the main text focused, this Appendix provides extended plots and tables that complement our core findings. It also summarizes uncertainty calibration (ECE, NLL, Brier) with reliability diagrams, computational complexity and runtime, and sensitivity to the number of GNP blocks and the co-attention temperature, together with exact splits, negative sampling, and seeds for reproducibility.

D.1 Additional results for the transductive setting

Fig. 6 visualizes ranking quality (ROC/PR curves) alongside thresholded Precision/Recall bars on DrugBank and Twosides. Across a wide recall range, `MPNP-DDI` maintains stronger precision than baselines, indicating more robust ranking of positive interactions. The bar plots provide a thresholded view aligned with validation selection, illustrating consistent gains in both precision and recall. Fig. 7 further contrasts `MPNP-DDI` and

GMPNN-CS on Twosides across multiple random seeds; the plot reports mean performance with uncertainty, and significance is assessed using a two-sided paired comparison across identical splits.



(a) DrugBank ROC/PR and P/R bars **(b)** Twosides ROC/PR and P/R bars

Figure 6. Full curve visualizations complementing Table 2.

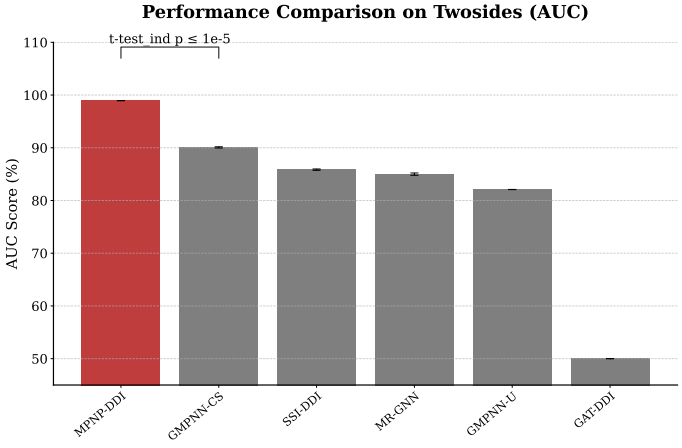


Figure 7. Statistical significance (Twosides): MPNP-DDI vs. GMPNN-CS.

D.2 Full inductive scalability table

Table 4 reports inductive performance as the training ratio increases. AUROC and AUPR improve monotonically with more supervision, indicating that the learned multi-scale representations transfer to unseen drugs. Thresholded metrics (F1/Precision/Recall) show mild non-monotonicity at intermediate ratios, a common effect of class imbalance and threshold selection; at full data (100%), all metrics reach their best levels with reduced variance across seeds.

Table 4. Inductive scalability (mean ± std, %). Ten independent seeds per ratio.

Training Ratio	Test AUROC	Test AUPR	Test F1	Test Precision	Test Recall
10%	51.20 ± 0.21	50.25 ± 0.30	50.95 ± 0.13	50.92 ± 0.22	50.99 ± 0.14
20%	51.66 ± 0.23	51.85 ± 0.22	52.96 ± 0.18	52.76 ± 0.38	53.15 ± 0.28
40%	57.27 ± 0.76	57.80 ± 0.31	47.33 ± 2.21	56.72 ± 1.47	40.75 ± 3.25
60%	62.89 ± 2.03	60.40 ± 2.12	59.97 ± 4.54	60.36 ± 2.16	60.80 ± 12.44
80%	69.27 ± 1.74	65.90 ± 2.44	58.53 ± 6.13	65.39 ± 2.83	58.06 ± 11.41
100%	75.12 ± 2.70	70.20 ± 2.11	68.29 ± 0.47	68.88 ± 2.96	67.92 ± 2.02