ARDiff: Anisotropic Residual Diffusion for Heterogeneous Graph Learning

Yong Chen[1*], Li Li[2*], Nannan Zong[3], Zhihui Liu[4], Song-Zhi Su[1, 3†]
[1]Institute of Artificial Intelligence, Xiamen University, China
[2]School of Artificial Intelligence, Shenzhen Polytechnic University, China
[3]School of Informatics, Xiamen University, China
[4]Xiamen Truesight Technology Co., Ltd., China
{ychen, beach}@stu.xmu.edu.cn, iamlylee@szpu.edu.cn, lzh@truesightai.com, ssz@xmu.edu.cn

Abstract

Learning representations on graphs is foundational for many downstream tasks, and its synergy with diffusion models has emerged as a promising direction. However, diffusion-based methods for heterogeneous graphs remain underexplored, confronting two principal challenges: (1) The presence of noise and structural heterogeneity in graphs makes it challenging to accurately capture semantic transitions among diverse relation types. (2) The isotropic Gaussian noise used in forward diffusion fails to reflect graphs' inherent semantics and structural anisotropy. To address these, we propose ARDiff, a novel framework that integrates residual diffusion with anisotropic noise for heterogeneous graph learning. Specifically, we propose a semantic residual diffusion mechanism that progressively refines node embeddings by orchestrating transitions from low-semantic (high-noise) to high-semantic (low-noise) relational contexts, thus enabling stepwise distillation of task-relevant information. In addition, to address the limitations of conventional diffusion, we introduce an anisotropic diffusion strategy: in the forward process, noise injection is oriented by structural and semantic priors; in the denoising step, a conditional diffusion mechanism is guided by a random walk encoding, enhancing both topological consistency and semantic alignment. Extensive evaluation on heterogeneous graph datasets demonstrates that ARDiff significantly surpasses current leading methods in link prediction and node classification, setting a new paradigm and benchmark in heterogeneous graph representation learning.

Code — https://github.com/s35lay/ARDiff

Introduction

Diffusion models (Croitoru et al. 2023) have achieved remarkable success in representation learning for computer vision tasks, such as image representation and semantic segmentation (He et al. 2025). Despite extensive studies in vision, diffusion models remain underexplored in graph representation learning, especially for heterogeneous graphs (Yang et al. 2023a). Their denoising mechanism aligns well with graph-structured data, where structural noise is prevalent and meaningful patterns are often obscured (Gasteiger, Weißenberger, and Günnemann 2019). Through iterative denoising, diffusion models hold promise for learning more expressive and robust graph representations.

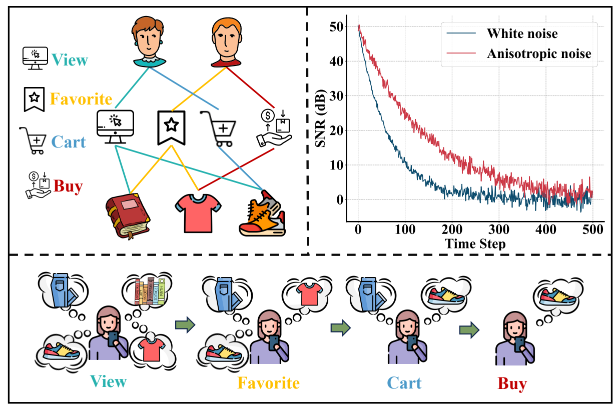

Figure 1: Illustration of Heterogeneous graphs: (1) Along specific relation chains, noisy interactions tend to diminish progressively, resembling the denoising in diffusion models. (2) Isotropic noise leads to a rapid decline in the SNR.

Heterogeneous graphs demonstrated effectiveness in various domains, including bibliographic networks (Yu et al. 2012), medical data (Kim et al. 2023), and recommender systems (Xiong et al. 2025). Unlike homogeneous graphs, they contain multiple types of nodes and edges, enabling richer semantic relations and more expressive structural representations, as shown in Figure 1. Heterogeneous graph neural networks (HGNNs) are widely adopted to model such data via relation-aware message passing (Bing et al. 2023). Early studies (Zhang et al. 2019; Yang et al. 2023b) show that leveraging structural and semantic heterogeneity significantly improves graph learning. Representative models like HGT (Tian et al. 2023) and HAN (Wang et al. 2019b) employ attention to capture the importance of different metapaths, enhancing both semantic-level and node-level representations. Recent advances in graph self-supervised learning further improve robustness against data sparsity and noise. For example, HeCo (Wang et al. 2021) proposes a contrastive learning framework that performs cross-view discrimination via meta-path masking. Meanwhile, generative methods aim to reconstruct node features and relations,

[*]These authors contributed equally.
[†]Corresponding author.

offering a promising direction to learn high-quality representations in heterogeneous graphs (Wang et al. 2025).

Despite recent advances, heterogeneous graph learning still faces two fundamental challenges. First, inherent noise in heterogeneous relations degrades the quality of the representation (Li et al. 2024). The aggregation and propagation mechanisms in HGNNs are sensitive to noisy nodes and edges, making it difficult to suppress irrelevant information. Existing graph self-supervised methods (Liu et al. 2022; Yang et al. 2022a) lack specialized mechanisms to handle complex noise that comes from various semantic types. Furthermore, some pretext tasks may introduce additional noise. Second, current methods struggle to model complex transition patterns among heterogeneous relations (Pang et al. 2022). Although attention (Yang et al. 2021) and meta-path-based approaches (Hang et al. 2024) capture some dependencies, they fail to represent fine-grained semantic changes. These limitations hinder the extraction of task-relevant semantics and degrade downstream performance.

To address these challenges, we propose a diffusion-based framework for heterogeneous graph learning. Consider the Tmall dataset, as illustrated in Figure 1, where user–item interactions are categorized into four relation types: *View*, *Favorite*, *Cart*, and *Buy*. These relations naturally form a semantic progression with decreasing noise levels. Users typically browse for many irrelevant items, indicate moderate interest by *Favorite*, demonstrate stronger intent by *Cart*, and finally confirm intent via *Buy*. This sequence reflects a transition from weak to strong semantics, and from high to low noisy interactions. Motivated by this observation, we introduce a semantic residual diffusion in latent space that can better handle the graph sparsity and discreteness. It combines diffusion with semantic chains of relations to eliminate noise and capture task-relevant semantic transitions, yielding high-quality heterogeneous graph representations.

Moreover, conventional diffusion models typically adopt an isotropic forward process, ignoring the structural and relational dependencies inherent in graphs. As shown in Figure 1, this leads to a rapid decline in the signal-to-noise ratio (SNR) and a misalignment with the anisotropic nature of heterogeneous graphs (Yang et al. 2023a; Xia et al. 2025). To overcome this limitation, we introduce an anisotropic diffusion strategy that explicitly integrates structural and semantic information into the noise modeling process. Specifically, we introduce a learnable orthogonal decomposition that links injected noise to graph topology and relation semantics, enabling the model to adaptively modulate noise direction and magnitude in a context-aware manner. To ensure consistency between the forward and reverse processes, we further introduce an information symmetry design. During the denoising phase, graph random walk features are incorporated as conditional input to preserve structural perception in residual learning between heterogeneous relations.

In summary, ARDiff consists of two core components: semantic residual diffusion and anisotropic diffusion. The main contributions can be summarized as follows:

- We propose a semantic residual diffusion mechanism in the latent space, which not only filters noise for each relation through the diffusion process, but also distills residual semantics between heterogeneous relation types to model complex semantic transitions effectively.
- We design an anisotropic diffusion strategy that jointly employs structural and semantic dependencies to learn an adaptive noise distribution in the forward diffusion process. In the reverse process, we employ a conditional denoising paradigm guided by graph-based random walks to enhance the learning of relational residuals.
- We conduct extensive experiments on diverse heterogeneous graph datasets, and the results consistently show that ARDiff outperforms existing baselines, demonstrating its robustness and modeling superiority.

Related Work

Graph Neural Networks. As a foundational paradigm for graph representation learning, GNNs achieve significant success in diverse domains, including recommender systems, knowledge graphs, and protein prediction (Zhou et al. 2020). Representative models like GAT (Veličković et al. 2017) and LightGCN (He et al. 2020) focus primarily on homogeneous or bipartite graphs, limiting their capacity to model complex heterogeneous semantics. To address this, we incorporate a diffusion model into the GNN paradigm, leveraging its denoising ability in latent space to capture semantic transitions across heterogeneous relations.

Heterogeneous Graph Learning. Heterogeneous graphs are prevalent in real-world applications (Kim et al. 2023; Xiong et al. 2025). To capture their rich semantic structures, HGNNs extend classical GNNs with relation-aware designs. Methods such as HetGNN (Zhang et al. 2019) and MAGNN (Fu et al. 2020) incorporate meta-path attention to aggregate information from semantical substructures, while HGT (Hu et al. 2020) adopts a transformer-based framework to model relation-aware interactions. LatGRL (Shen and Kang 2025) further constructs latent fine-grained subgraphs to alleviate node heterophily. DMGI (Park et al. 2020) maximizes mutual information between local and global views. Despite their effectiveness, these methods typically overlook the impact of semantic noise and irrelevant context. In contrast, our method introduces a latent-space diffusion mechanism that progressively denoises heterogeneous signals and captures residual semantics across relations, leading to more robust and task-relevant representations.

Generative Models for Graph Learning. Graph generation aims to learn the underlying distribution of observed graphs and generate new topologies (Wang et al. 2025). Early methods (ERDdS and R&wi 1959; Kleinberg 2000) relied on specific statistical features, while recent advances adopt deep generative models such as variational autoencoders (Jin, Barzilay, and Jaakkola 2018), generative adversarial networks (Tang et al. 2024), and diffusion models (Kong et al. 2023). For instance, GraphVAE (Simonovsky and Komodakis 2018) employs latent variables to model graph structures. GraphGAN (Wang et al. 2018) leverages adversarial training for structure generation. RecDiff (Li, Xia, and Huang 2024) applies a diffusion model with multi-view control in social recommendation. However, these models are limited to graph denoising and semantic modeling. Our ARDiff employs a diffusion framework to filter noise and capture relational semantics, further incorporating graph anisotropy for enhanced representation learning.

Preliminaries

Heterogeneous graph is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ represent the node set and the edge set, respectively. Each node $v_i \in \mathcal{V}$ and each edge $(v_i, v_j) \in \mathcal{E}$ are associated with their mapping functions $\psi(v) : \mathcal{V} \rightarrow \mathcal{A}$ and $\phi(v_i, v_j) : \mathcal{E} \rightarrow \mathcal{R}$. Here, $\mathcal{A}$ and $\mathcal{R}$ denote the node types and edge types, respectively. For convenience, a heterogeneous graph is recorded by the binary adjacency matrix $\mathbf{A}$ of size $|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|$. Each element $a^r_{i,j} \in \mathbf{A}$ equals 1, if there is a edge of type $r \in \mathcal{R}$ between nodes v_i and v_j, otherwise $a^r_{i,j} = 0$.

A predictive model f for heterogeneous graphs can be divided into an encoding phase and a prediction phase, formally as: $f(\mathcal{G}) = \mathrm{Enc}(\mathcal{G}) \circ \mathrm{Pred}(\mathbf{E})$. The encoding phase $\mathrm{Enc}(\mathcal{G})$ learns a function $\Psi : \mathcal{V} \rightarrow \mathbb{R}^d$ that embeds $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$ for nodes $v_i \in \mathcal{V}$. while the prediction phase $\mathrm{Pred}(\mathbf{E})$ generates task-specific predictions based on the learned embeddings. For node classification, $\mathrm{Pred}(\mathbf{E})$ employs a multilayer perceptron (MLP) to infer the classes of the nodes. For link prediction, the dot-product operator is utilized to infer the existence of edges in the $\mathrm{Pred}(\mathbf{E})$.

Methodology

In this section, we introduce our proposed ARDiff method and present the overall framework in Figure 2.

Semantic Residual Diffusion

In heterogeneous graphs, distinct relation types often carry varying levels of semantic intensity that evolve in a cascading manner. Taking the Tmall dataset as an example, the relations of user interactions (*View*, *Favorite*, *Cart*, and *Buy*) can be naturally ordered by increasing intention and engagement. This reflects a semantic trajectory in which weaker exploratory signals (e.g., view) progressively transition to stronger and more definitive actions (e.g., purchase). As the semantic strength increases along this chain, the noise in the relational graph correspondingly decreases. To capture this transition, we define Chain-of-Semantics (CoS), which models the ordered evolution of semantics across relation types. Formally, CoS can be expressed as follows:

$$r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_K, \quad \forall r_i \in \mathcal{R}, 1 \leq i \leq K. \qquad (1)$$

The CoS, organized from high-noise to low-noise relations, inherently aligns with the noising–denoising paradigm in diffusion models. Therefore, we employ CoS as the foundational structure of our semantic residual diffusion framework to distill relation-specific semantics and gradually improve the representation by reducing relation noise.

To facilitate the diffusion process, we operate in the latent space of the graph, where representations are less affected by structural discreteness and sparsity. Specifically, we employ an encoder $\mathrm{Enc}(\mathcal{G})$ to obtain the node embeddings. Inspired by the simplicity and effectiveness of GCNs, we adopt a GCN-based architecture for the encoder in ARDiff. The encoding process is defined as follows:

$$\mathbf{E}_r = \sum_{l=0}^{L} \mathbf{E}^l_r, \;\; \mathbf{E}^l_r = \mathrm{N}(\sigma(\mathbf{D}^{-1/2}\mathbf{A}_r\mathbf{D}^{-1/2}\mathbf{E}^{l-1}_r)), \quad (2)$$

where where $\mathbf{A}_r$ denotes the adjacency matrix for each heterogeneous relation $r \in \mathcal{R}$, and $\mathbf{D}$ denotes its corresponding diagonal degree matrix. $\mathbf{E}^l_r \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the embedding matrix at the l-th GCN layer. Through L_2 normalization $\mathrm{N}(\cdot)$, activation function $\sigma(\cdot)$, and L iterations, the final node embeddings $\mathbf{E}_r$ of each relation type are generated.

Our diffusion aims to iteratively distill the semantic residuals between the high-noise relations r_i and adjacent low-noise counterparts r_{i+1} along CoS, capturing the transition dynamics. We conduct this diffusion process in the latent representation space of the heterogeneous graph data to achieve the following step-by-step transformation:

$$\mathbf{E}_{r_1} \xrightarrow{\varphi} \tilde{\mathbf{E}}_{r_1} \xrightarrow{\varphi'} \hat{\mathbf{E}}^{res}_{r_{1,2}} \xrightarrow{\zeta} \hat{\mathbf{E}}_{r_2} \xrightarrow{\varphi} \ldots \xrightarrow{\zeta} \hat{\mathbf{E}}_{r_K}. \quad (3)$$

Here, φ denotes the forward diffusion, which injects noise into $\mathbf{E}_r$, while φ' represents the reverse process, which denoises the representation. $\hat{\mathbf{E}}^{res}_{r_{1,2}}$ refers to the semantic residual between $\mathbf{E}_{r_1}$ and $\mathbf{E}_{r_2}$. These residuals are propagated and accumulated along CoS, progressively enhancing node representations by filtering out noise and distilling semantic features. ζ is the weighted sum function to obtain the next low-noise relation embeddings in CoS, which is defined as:

$$\hat{\mathbf{E}}_{r_{i+1}} = \gamma_i(\mathbf{E}_{r_i} + \hat{\mathbf{E}}^{res}_{r_{i,i+1}}) + (1 - \gamma_i)\mathbf{E}_{r_{i+1}}, \qquad (4)$$

where γ_i is a learnable balance factor for the relation r_i to avoid error accumulation when learning residuals along CoS. The specific diffusion process for calculating semantic residuals $\hat{\mathbf{E}}^{res}_{r_{i,i+1}}$ will be detailed in the following sections.

By focusing on modeling semantic differences instead of fully reconstructing node representations, the reverse process simplifies the learning objective. This differential modeling strategy facilitates the capture of behavioral transition patterns between relations while reducing computational overhead. As a result, ARDiff achieves greater efficiency and accuracy in recovering meaningful relational semantics and structure from noisy heterogeneous graphs. This, in turn, benefits downstream tasks by providing more reliable and informative representations.

Anisotropic Diffusion

Forward Process. In heterogeneous graph learning, the isotropic noise in vanilla diffusion blurs relational distinctions and causes a rapid decline in SNR. To this end, we introduce a novel anisotropic noise method that transforms initial isotropic Gaussian noise into anisotropic noise by integrating the graph's semantic and structural information. The process $q(\mathbf{H}_t|\mathbf{H}_{t-1})$ starts with the encoded latent representations $\mathbf{H}_0 = \mathbf{E}_{r_1}$, and increases the noise as follows:

$$\mathbf{H}_t = \sqrt{\bar{\alpha}_t}\mathbf{H}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}', \qquad (5)$$

$$\boldsymbol{\epsilon}' = w_1\bar{\boldsymbol{\epsilon}}_{\parallel} + w_2\bar{\boldsymbol{\epsilon}}_{\perp}, \qquad (6)$$

$$\bar{\boldsymbol{\epsilon}} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \qquad (7)$$

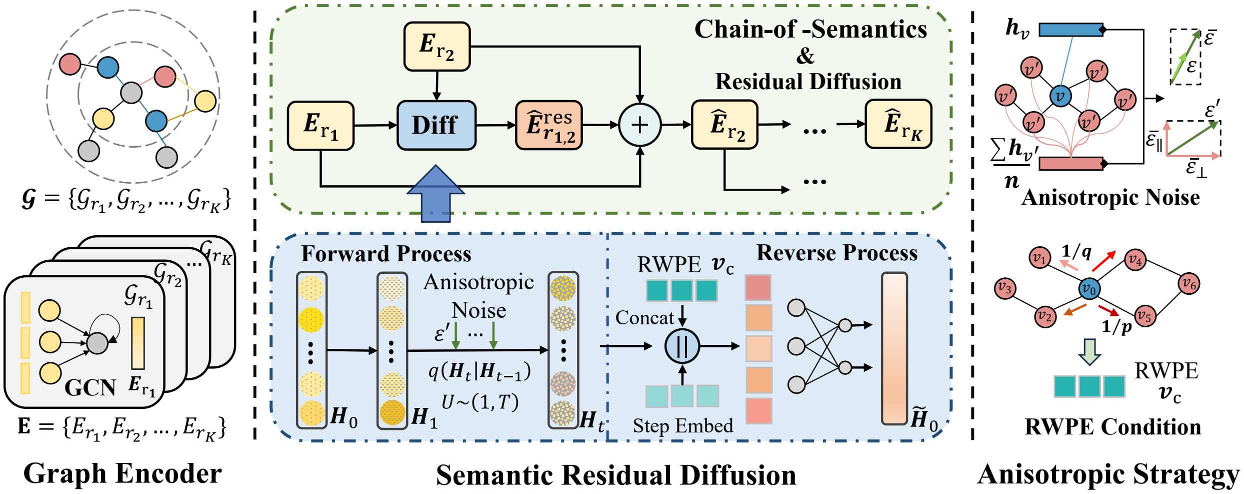

Figure 2: Overall framework of our ARDiff: (1) GCN-based encoder maps the graph into embeddings based on relation types. (2) Semantic residual diffusion models semantic transitions between relations along a predefined chain-of-semantics, progressively filtering noise and distilling task-relevant signals. (3) Anisotropic strategy incorporates anisotropic noise in the forward process and RWPE-based conditional diffusion in the reverse process to enhance graph representation learning.

The parameter $\bar{\alpha}_t := \prod_{i=0}^{t}(1-\beta_i) \in (0,1)$ represents the fixed variance schedule by a decreasing sequence $\{\beta_{1:T} \in (0,1)\}$. We impose two additional constraints for anisotropic noise conditioned on structural and semantic factors. Specifically, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ in Equation (7) represent the mean and variance of each node's k-order neighborhood in $\mathbf{H}_0$. Equation (6) decomposes the noise into orthogonal components $\bar{\boldsymbol{\epsilon}}_{\|}$ and $\bar{\boldsymbol{\epsilon}}_{\perp}$ along the parallel and perpendicular directions to the $\boldsymbol{\mu}-\mathbf{H}_0$ axis: the former preserves local coherence, while the latter encourages latent space exploration. Two components are balanced by weights w_1 and w_2 defined as follows:

$$[w_1, w_2] = \mathrm{softmax}\left(\mathrm{MLP}(\bar{\boldsymbol{\epsilon}}_{\perp} \| \bar{\boldsymbol{\epsilon}}_{\|})\right), \tag{8}$$

where $\|$ denotes concatenation. Consequently, the combined constraints produce noise $\boldsymbol{\epsilon}'$ for the forward process, preserving inherent data structure and slowing SNR decay to facilitate effective feature extraction.

Reverse Process. The reverse process in ARDiff aims to reconstruct relational residuals in latent space from noisy heterogeneous representations $\mathbf{H}_t$, thereby learning transitions between noisy semantic relations. As the forward process is structure-aware, we incorporate the graph structure as a condition $\mathbf{c}_r$ during denoising, following the conditional diffusion paradigm $p(\mathbf{H}_{t-1}|\mathbf{H}_t, \mathbf{c}_r)$. Specifically, the mean $\mu_\theta(\cdot)$ of the denoising distribution is predicted using a two-layer MLP with parameter θ, as defined below:

$$\mu_\theta(\mathbf{h}_t, t, \mathbf{c}) = \mathrm{MLP}^2\left(\mathbf{h}_t \| \mathbf{s}_t \| \mathbf{v}_{\mathbf{c}}\right). \tag{9}$$

We concatenate the conditional feature $\mathbf{v}_{\mathbf{c}}$, the time step-specific embedding $\mathbf{s}_t$, and the t-step hidden vector $\mathbf{h}_t$ as the input of $\mathrm{MLP}^2(\cdot)$. To extract graph-structure features, we employ Random Walk Positional Encoding (RWPE) as $\mathbf{c}_r$, which encodes each node by its return probability within k-step random walks. As a conditioning signal, RWPE captures both topological position and connection strength, enabling the model to distinguish structural patterns from local to global scales. The k-step RWPE is defined as follows:

$$\mathbf{c}_{r,i} = \left[(\bar{\mathbf{A}}_r)_{ii}, (\bar{\mathbf{A}}_r^2)_{ii}, \ldots, (\bar{\mathbf{A}}_r^k)_{ii}\right], \; \bar{\mathbf{A}}_r = \mathbf{A}_r\mathbf{D}^{-1}, \tag{10}$$

where $\mathbf{c}_r \in \mathbb{R}^{|V| \times k}$ and combine the learnable parameter matrices $\mathbf{w}$ and $\mathbf{b}$ with the activate function σ to model the structural transfer vector $\mathbf{v}_{\mathbf{c}}$ as follows:

$$\mathbf{v}_{\mathbf{c}} = \sigma\left(\mathbf{w}(\mathbf{c}_{r_{i+1}} - \mathbf{c}_{r_i}) + \mathbf{b}\right). \tag{11}$$

Being structure-sensitive and free from sign ambiguity, RWPE provides robust guidance for the reverse process. Conditioned on this graph-aware signal, ARDiff can perceive relational differences and accurately recover fine-grained residuals step by step along the CoS.

Model Training for ARDiff

Diffusion Loss Function. To optimize the diffusion module, we model the diffusion process between each pair of relations (r_i, r_{i+1}) by leveraging the residual embedding $\mathbf{E}^{res}_{r_{i,i+1}} = \mathbf{E}_{r_{i+1}} - \hat{\mathbf{E}}_{r_i}$, where $\hat{\mathbf{E}}_{r_1} = \mathbf{E}_{r_1}$. This residual captures the semantic transition from a high-noise relation to a low-noise one. Serving as a proxy for the Evidence Lower Bound (ELBO), it guides the optimization process in the latent space. The ELBO-based objective over the residual representation $\mathbf{E}^{res}_{r_{i,i+1}}$ is formulated as follows:

$$\log p(\mathbf{h}_0') \geq \underbrace{\mathbb{E}_{q(\mathbf{h}_1|\mathbf{h}_0)}\left[\log p_\theta\left(\mathbf{h}_0'|\mathbf{h}_1\right)\right]}_{\text{(reconstruction term)}} \tag{12}$$

$$- \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{h}_t|\mathbf{h}_0)}\left[D_{\mathrm{KL}}\left(q\left(\mathbf{h}_{t-1}|\mathbf{h}_t, \mathbf{h}_0\right) \| p_\theta\left(\mathbf{h}_{t-1}|\mathbf{h}_t, \mathbf{c}_r\right)\right)\right]}_{\text{(denoising comparison term)}},$$

where $\mathbf{h}_0' \in \mathbf{E}_{r_{i,i+1}}^{res}$ and it consists of two optimization terms. The denoising comparison term encourages the conditional distribution $p_\theta(\mathbf{h}_{t-1}|\mathbf{h}_t, \mathbf{c}_r)$ along the diffusion trajectory to align with the true distribution $q(\mathbf{h}_{t-1}|\mathbf{h}_t, \mathbf{h}_0)$, minimizing the KL divergence between them. Following previous studies (Li et al. 2025), we simplify the standard deviation as a time-dependent scalar $\sigma^2(t)\mathbf{I}$. This enables the denoising loss at the timestep t to be transformed as:

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{h}_t|\mathbf{h}_0)}\left[\delta_t \|\hat{\mathbf{h}}_\theta(\mathbf{h}_t, t, \mathbf{c}_r) - \mathbf{h}_0'\|^2\right], \quad (13)$$

where $\delta_t = \big(\bar{\alpha}_{t-1}/(1-\bar{\alpha}_{t-1}) - \bar{\alpha}_t/(1-\bar{\alpha}_t)\big)/2$ denotes dynamical changes in SNR over time t. Finally, we uniformly sample the diffusion timestep $t \sim \mathcal{U}(1, t)$, and define the diffusion loss as $\mathcal{L}_{\text{diff}} = \mathbb{E}_{t\sim\mathcal{U}(1,t)}\mathcal{L}_t$.

Prediction and Optimization. For the link prediction, following Equations (3) and (4), ARDiff performs semantic residual diffusion along the CoS, progressively refining and producing the final embedding $\mathbf{E}_{r_k}$ for prediction. We adopt the BPR loss to optimize the predictions $\hat{y}$ as follows:

$$\mathcal{L}_{\text{main}} = -\sum_{u,v^+,v^-} \log\big(\text{sigmoid}(\hat{y}_{u,v^+} - \hat{y}_{u,v^-})\big), \quad (14)$$

where $\hat{y}_{u,v} = \tilde{\mathbf{e}}_u^\top \tilde{\mathbf{e}}_v$ and (u, v^+, v^-) represent positive and negative sample triplets. For the node classification, we employ the cross entropy loss as follows:

$$\mathcal{L}_{\text{main}} = -\sum_{v_i\in\mathcal{V}} \log\big(\text{softmax}(\text{MLP}(\tilde{\mathbf{e}}_i))_{y_i}\big). \quad (15)$$

Experiments

We evaluate the effectiveness of ARDiff through extensive experiments, following research questions (RQs):

- **RQ1:** How does ARDiff perform on link prediction and node classification compared to baseline models?
- **RQ2:** What are the contributions of the key components in ARDiff to its overall performance?
- **RQ3:** How do the different settings of key hyperparameters impact performance?
- **RQ4:** Can ARDiff offer interpretable insights in specific cases through visualization?

Experimental Settings

Datasets. To validate the effectiveness of ARDiff, we conduct extensive experiments on link prediction and node classification tasks across five representative datasets. Specifically, link prediction is evaluated on three heterogeneous behavior datasets: **Retailrocket** (Yang et al. 2022b), **IJCAI** (Du, Yu, and Wang 2025), and **Tmall** (Xia et al. 2020), while node classification is assessed via meta-path relationships on two academic network datasets: **DBLP** (Kumar et al. 2025) and **Aminer** (Wan et al. 2019). The detailed statistics of all datasets are summarized in Table 1.

Link Prediction Task

Dataset	User #	Item #	Link #	Interaction Types
Retail Rocket	2174	30113	97,381	View, Cart, Transaction
IJCAI	17435	35920	799,368	View, Favorite, Cart, Purchase
Tmall	31882	31232	1,451,29	View, Favorite, Cart, Purchase

Node Classification Task

	Node	Metapath		Node	Metapath
DBLP	Author:4057 Paper:14328 Conference:20 Term:7723	APA APCPA APTPA	AMiner	Paper:6564 Author:13329 Reference:35890	PAP PRP POS

Table 1: Statistics of experimental datasets.

Baseline Methods. We compare 18 baseline methods to comprehensively evaluate ARDiff, which can be grouped into four categories. (1) Non-Graph Neural Networks: BPR (Rendle et al. 2009). (2) Recommendation Models with Heterogeneous Relations: NMTR (Gao et al. 2019), MBGCN (Jin et al. 2020), and MATN (Xia et al. 2020). (3) Homogeneous GNNs-based methods: GAE (Kipf and Welling 2016), GraphSage (Hamilton, Ying, and Leskovec 2017), PinSAGE (Ying et al. 2018), DGI (Veli et al. 2019), NGCF (Wang et al. 2019a), and GraphMAE (Hou et al. 2022). (4) Heterogeneous GNNs-based methods: Mp2vec (Dong, Chawla, and Swami 2017), HERec (Shi et al. 2018), HetGNN (Zhang et al. 2019), HAN (Wang et al. 2019b), HGT (Hu et al. 2020), DMGI (Park et al. 2020), and HeCo (Wang et al. 2021). In particular, we compare ARDiff with DiffGraph (Li et al. 2025), the latest model integrating GNN and diffusion, to highlight advances in modeling complex semantics.

Evaluation Protocols. We follow the same evaluation protocols for ARDiff and all baselines as detailed below:

- Link Prediction: We adopt the leave-one-out strategy, using users' last interactions under the purchasing behavior for testing. Evaluation is based on Recall@20 and NDCG@20, standard metrics in ranking tasks.
- Node Classification: We train the model using 20, 40, and 60 labeled nodes per class, and evaluate it on 1,000 nodes each for validation and testing. Performance is measured using Micro-F1, Macro-F1, and AUC.

Hyperparameter Settings. We implement ARDiff using PyTorch and optimize it with the Adam algorithm. The hyperparameter settings are as follows:

- Link Prediction: We tune the learning rate from $1e^{-3}$ to $1e^{-4}$, GCN layers in $\{1, 2, 3, 4\}$, embedding dimensions from 64 to 512, and conditional vector dimensions from 16 to 128. For diffusion, the noise steps range from 0 to 300 and the noise scales from $1e^{-1}$ to $1e^{-6}$. The batch size is selected between 512 and 8192.
- Node Classification: For better comparison, we follow the settings of HeCo and DiffGraph. We set the hidden embedding to 64, search for the learning rate from $1e^{-2}$ to $1e^{-4}$, and select the GCN layers from 1 to 9.

Link Prediction Performance on Public Datasets (p-value$< 1e^{-7}$)																				
Dataset	BPR		PinSAGE		NGCF		NMTR		MBGCN		HGT		MATN		DiffGraph		**ARDiff**		**Improve**	
	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
Rocket	0.0308	0.0235	0.0425	0.0249	0.0407	0.0254	0.0463	0.0265	0.0503	0.0264	0.0415	0.0253	0.0528	0.0306	0.0585	0.0329	**0.0755**	**0.0472**	**29.06%**	**43.47%**
IJCAI	0.0056	0.0036	0.0104	0.0046	0.0094	0.0032	0.0107	0.0053	0.0114	0.0045	0.0125	0.0054	0.0134	0.0053	0.0142	0.0061	**0.0185**	**0.0082**	**30.28%**	**34.43%**
Tmall	0.0251	0.0132	0.0370	0.0155	0.0397	0.0172	0.0440	0.0194	0.0422	0.0179	0.0434	0.0194	0.0466	0.0201	0.0553	0.0246	**0.0667**	**0.0313**	**20.61%**	**27.24%**

Node Classification Performance on Public Datasets														
Dataset	Metric		GraphSage	GAE	Mp2vec	HERec	HetGNN	HAN	DGI	DMGI	HeCo	GraphMAE	DiffGraph	**ARDiff**
DBLP	Micro-F1	20	70.72±7.8	91.42±0.1	89.49±0.2	90.01±0.5	89.91±0.8	90.02±0.9	89.16±2.1	90.67±0.3	91.74±0.3	89.11±0.5	91.30±0.3	**93.95±0.1**
		40	72.21±8.3	90.21±0.4	89.27±0.2	90.15±0.3	89.11±0.6	89.41±0.8	89.24±0.4	89.64±0.5	90.76±0.3	87.81±0.6	90.16±0.2	**93.47±0.2**
		60	73.54±7.1	90.65±0.3	89.07±0.2	91.10±0.3	90.41±0.5	90.26±0.7	90.36±0.8	90.67±0.4	91.4±0.2	89.81±0.5	90.72±0.5	**94.35±0.2**
	Macro-F1	20	71.94±8.1	90.95±0.1	88.67±0.2	89.74±0.3	90.01±1.1	89.16±0.8	87.96±2.1	89.67±0.3	91.24±0.1	88.01±0.8	90.75±0.2	**93.56±0.3**
		40	73.44±8.2	89.55±0.2	88.67±0.2	90.01±0.2	88.61±0.7	88.86±1.1	88.56±0.5	89.27±0.7	90.24±0.4	86.71±0.6	89.72±0.3	**93.15±0.3**
		60	73.74±7.4	90.05±0.2	90.27±0.1	90.14±0.3	89.61±0.5	89.16±0.9	89.56±0.6	90.67±0.2	88.14±0.8	93.11±0.2	90.01±0.3	**93.63±0.2**
	AUC	20	90.44±4.1	98.15±0.1	97.70±0.1	98.24±0.3	97.91±0.3	98.06±0.6	96.93±1.1	97.63±0.2	98.24±0.1	92.22±3.1	98.49±0.1	**99.13±0.1**
		40	91.44±3.7	97.75±0.2	97.07±0.1	97.94±0.1	97.71±0.3	97.46±0.5	97.16±0.4	97.27±0.3	98.04±0.1	91.71±2.5	98.37± 0.1	**99.05±0.1**
		60	91.74±3.5	98.45±0.2	97.87±0.1	98.44±0.1	97.91±0.1	97.76±0.5	97.66±0.5	97.67±0.4	98.54±0.1	91.61±2.4	98.65± 0.1	**99.39±0.1**
AMiner	Micro-F1	20	49.54±3.1	65.68±2.1	60.87±0.3	63.74±1.3	61.41±2.2	68.86±4.8	61.96±3.7	62.67±3.3	78.84±1.7	68.01±0.3	76.46±0.9	**82.43±0.6**
		40	53.44±2.2	71.55±1.2	69.24±0.5	71.37±0.6	68.42±2.7	77.15±1.6	63.77±2.6	63.12±2.6	80.24±0.6	74.21±0.2	81.24±1.1	**85.14±1.0**
		60	51.74±2.3	67.75±1.8	63.87±0.5	69.74±0.8	65.61±2.5	73.76±1.9	63.56±3.4	62.67±2.4	82.34±1.6	72.21±0.2	80.09±1.0	**83.22±1.2**
	Macro-F1	20	41.94±2.5	60.95±2.2	53.67±0.5	59.14±1.3	50.05±1.0	55.78±3.2	51.96±3.1	59.67±2.4	71.34±1.1	62.51±0.2	67.66±0.4	**73.72±0.8**
		40	44.64±1.2	65.55±1.6	64.67±0.5	65.51±0.7	58.61±0.7	63.86±1.5	54.76±2.6	61.87±2.0	73.74±0.5	68.11±0.2	73.59±0.9	**76.85±0.9**
		60	43.74±1.8	63.76±1.5	60.22±0.3	65.54±0.7	57.61±1.5	62.06±1.1	55.56±2.6	60.67±2.3	75.14±1.8	68.21±0.2	74.02±0.6	**75.87±0.7**
	AUC	20	70.83±2.1	84.75±1.0	81.20±0.3	83.34±0.5	77.91±1.3	78.86±2.6	75.89±2.1	86.33±0.8	90.65±0.5	85.44±4.2	88.62±0.5	**92.41±0.2**
		40	73.84±1.7	88.25±1.1	88.87±0.2	87.94±0.5	83.11±1.3	80.76±2.5	77.86±2.2	88.07±1.3	92.14±0.5	90.00±0.1	92.31±0.5	**94.72±0.3**
		60	74.14±1.5	87.05±0.9	85.57±0.1	87.75±0.5	84.71±1.0	80.36±1.5	77.26±1.5	87.17±1.7	92.54±0.8	88.31±0.1	92.46±0.6	**94.45±0.5**

Table 2: Overall performance comparison on the link prediction and node classification tasks.

Overall Performance Comparison (RQ1)

The performance of ARDiff and the baselines are detailed in Table 2, where we can obtain the following observations.

Outstanding Performance of ARDiff. Our ARDiff consistently outperforms all baselines in both link prediction and node classification tasks, demonstrating its superior graph learning capability. This advantage stems from its semantic residual diffusion along the CoS, which incrementally distills task-relevant semantics from heterogeneous relations. By modeling the dynamic semantic transitions between relation types, ARDiff effectively captures rich heterogeneity and enhances task-specific representations.

Impact of Heterogeneous Relations. The results indicate that models based on heterogeneous graph learning generally surpass those built on conventional homogeneous GNNs, underscoring the importance of exploiting relation-specific semantics. Such semantics not only enrich structural and relational representations, but also capture diverse interaction patterns that are often lost in homogeneous settings. This leads to more expressive node embeddings and ultimately boosts predictive performance in downstream tasks.

Effectiveness of Diffusion in Graph Learning. ARDiff outperforms state-of-the-art heterogeneous graph learning models by effectively leveraging graph semantics and structure to reconstruct graph representations through diffusion-based denoising. The semantic diffusion along the CoS captures nuanced variations across heterogeneous relations, while anisotropy strategies further enrich graph structural information, jointly enhancing the capacity of ARDiff for expressive and robust graph representation learning.

Ablation Study (RQ2)

To study the impact of model components, we individually remove the key modules from ARDiff to divide into four ablated variants: (1) **-D:** Remove the diffusion module. (2) **-R:** Replace semantic residual diffusion along CoS with vanilla diffusion. (3) **-A:** Remove anisotropy of noise. (4) **-C:** Remove the RWPE condition in the denoising process.

We conduct an ablation study on the link prediction task, and the results in Figure 3 reveal the following insights:

- Removing the diffusion module (-D) results in a significant performance drop, as the model loses its capacity to denoise relational noise and reconstruct meaningful graph semantics. This absence hampers the extraction of stable and task-relevant features, leading to degraded representations. These results highlight the effectiveness of ARDiff in diffusing within the graph semantic space.
- Replacement of semantic residual diffusion with vanilla diffusion (-R) compromises the modeling of semantic progression across heterogeneous relations in CoS. Unlike the residual approach, vanilla diffusion neglects the inherent order in relational semantics, leading to indiscriminate information propagation and a reduction in the

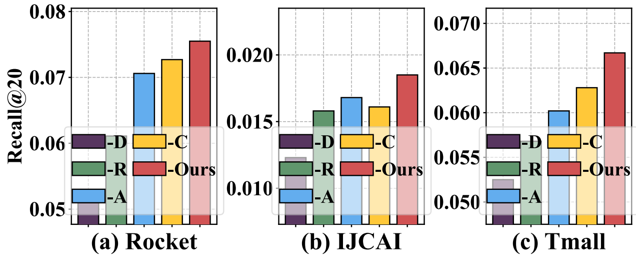

Figure 3: Module ablation study for ARDiff.

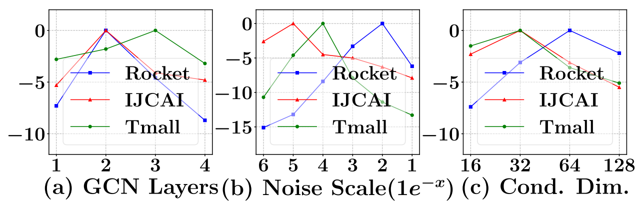

Figure 4: Hyperparameter study in Recall@20 changes (%).

discriminative capacity of node embeddings.

- Eliminating anisotropic noise injection (-A) affects the model's ability to adaptively regulate noise according to structural and semantic contexts. The anisotropic mechanism modulates both the magnitude and direction of noise in a context-aware manner, facilitating targeted denoising. Its absence leads to non-directional noise interference and destabilized learning dynamics.
- The exclusion of the RWPE condition (-C) during denoising weakens the model's ability to perceive the node positions and structural context. As a structural prior, RWPE guides the preservation of local topological patterns. Without it, the diffusion process is more likely to disrupt inherent graph structures, undermining the spatial consistency of the learned representations.

Effect of Hyperparameters (RQ3)

In this section, we investigate how crucial hyperparameters in three modules impact performance. The results, shown in Figure 4, are analyzed as follows:

- **Layers in GCN.** We search for the optimal setting within the range of 1 to 4 layers. Performance improves with increasing GCN layers up to a point, but declines thereafter due to noise accumulation and over smoothing. The optimal setting at 2 layers suggests that shallow networks strike a better balance by preserving semantic distinctions while effectively capturing structural signals.
- **Noise scale in diffusion.** The moderate noise scale promotes effective denoising and generalization, while excessive noise disrupts semantic integrity and degrades model performance. As it directly modulates residual strength, the noise scale is pivotal to diffusion stability. We tune it within $\{1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}\}$, where optimal noise scales improve model performance by enhancing denoising effectiveness.

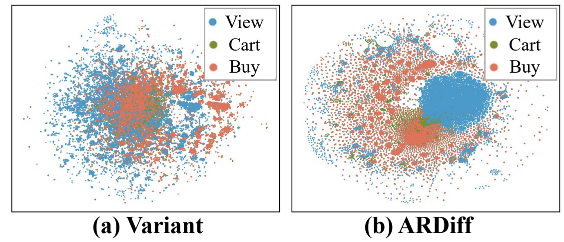

Figure 5: Embedding visualization on the Rocket dataset.

- **RWPE feature dimension in condition.** In the conditional denoising process, we explore the dimensions of RWPE features in 8, 16, 32, 64. Higher dimensions provide finer-grained positional encoding, enhancing structural modeling. While richer encodings capture subtle positional differences, overly high dimensions may introduce redundancy and noise, which leads to offset the benefits of positional features.

Case Study (RQ4)

To intuitively assess the representation capacity of ARDiff, we apply t-SNE to visualize node embeddings from the Retailrocket dataset, using color to distinguish different relation types, as shown in Figure 5. We compare ARDiff with a variant without semantic residual diffusion and anisotropic strategy. The visualization reveals that ARDiff produces more clearly separated clusters, with embeddings of similar semantic types more compactly grouped. This demonstrates that ARDiff effectively denoises and captures essential semantic transitions via residual-conditioned diffusion with anisotropic noise, resulting in more accurate and coherent representations. In contrast, the variant exhibits blurred cluster boundaries and overlapping regions, indicating inferior semantic disentanglement. This highlights the critical role of both semantic diffusion and anisotropy in modeling relational diversity. Furthermore, by progressively optimizing semantic flow along the CoS, ARDiff integrates and amplifies cross-relation signals, enhancing the expressiveness and discriminative power of graph representation.

Conclusion

This paper proposed ARDiff, a novel heterogeneous graph diffusion model that advances graph learning through two key innovations: (1) Anisotropic Residual Diffusion, which progressively denoises and learns semantic transitions between heterogeneous relations, effectively distilling task-relevant signals. (2) Anisotropic Strategy, enabling adaptive noise injection and denoising during the diffusion process, which perceptively incorporates graph structure and semantic information for more refined heterogeneous graph modeling. Extensive experiments on public benchmark datasets demonstrate ARDiff's superiority in both prediction accuracy and model robustness. Future work can explore extending ARDiff to temporal heterogeneous graphs, leveraging temporal information to enhance the diffusion process along the CoS for dynamic graph structures.

References

Bing, R.; Yuan, G.; Zhu, M.; Meng, F.; Ma, H.; and Qiao, S. 2023. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artificial Intelligence Review*, 56(8): 8003–8042.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9): 10850–10869.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 135–144.

Du, Y.; Yu, Z.; and Wang, H. 2025. Multi-behavior aware recommendation with joint contrastive learning and reinforced negative sampling. *Complex & Intelligent Systems*, 11(8): 352.

ERDdS, P.; and R&wi, A. 1959. On random graphs I. *Publ. math. debrecen*, 6(290-297): 18.

Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, 2331–2341.

Gao, C.; He, X.; Gan, D.; Chen, X.; Feng, F.; Li, Y.; Chua, T.-S.; and Jin, D. 2019. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th international conference on data engineering (ICDE)*, 1554–1557. IEEE.

Gasteiger, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion improves graph learning. *Advances in neural information processing systems*, 32.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hang, J.; Hong, Z.; Feng, X.; Wang, G.; Yang, G.; Li, F.; Song, X.; and Zhang, D. 2024. Paths2pair: Meta-path based link prediction in billion-scale commercial heterogeneous graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5082–5092.

He, C.; Shen, Y.; Fang, C.; Xiao, F.; Tang, L.; Zhang, Y.; Zuo, W.; Guo, Z.; and Li, X. 2025. Diffusion models in low-level vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 594–604.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, 2704–2710.

Jin, B.; Gao, C.; He, X.; Jin, D.; and Li, Y. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 659–668.

Jin, W.; Barzilay, R.; and Jaakkola, T. 2018. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, 2323–2332. PMLR.

Kim, S.; Lee, N.; Lee, J.; Hyun, D.; and Park, C. 2023. Heterogeneous graph learning for multi-modal medical data analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 5141–5150.

Kipf, T. N.; and Welling, M. 2016. Variational graph autoencoders. *arXiv preprint arXiv:1611.07308*.

Kleinberg, J. 2000. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 163–170.

Kong, L.; Cui, J.; Sun, H.; Zhuang, Y.; Prakash, B. A.; and Zhang, C. 2023. Autoregressive diffusion model for graph generation. In *International conference on machine learning*, 17391–17408. PMLR.

Kumar, C.; Dar, S. A.; Batchu, C.; Singamaneni, K. S.; and Panguluri, V. L. A. 2025. Peer collaboration in DBLP using graph convolutional network. *SN Computer Science*, 6(1): 71.

Li, M.; Liu, K.; Liu, H.; Zhao, Z.; Ward, T. E.; and Wu, X. 2024. Heterogeneous meta-path graph learning for higher-order social recommendation. *ACM Transactions on Knowledge Discovery from Data*, 18(8): 1–25.

Li, Z.; Xia, L.; Hua, H.; Zhang, S.; Wang, S.; and Huang, C. 2025. DiffGraph: Heterogeneous Graph Diffusion Model. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, 40–49.

Li, Z.; Xia, L.; and Huang, C. 2024. Recdiff: Diffusion model for social recommendation. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, 1346–1355.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. S. 2022. Graph self-supervised learning: A survey. *IEEE transactions on knowledge and data engineering*, 35(6): 5879–5900.

Pang, Y.; Wu, L.; Shen, Q.; Zhang, Y.; Wei, Z.; Xu, F.; Chang, E.; Long, B.; and Pei, J. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 775–783.

Park, C.; Kim, D.; Han, J.; and Yu, H. 2020. Unsupervised attributed multiplex network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5371–5378.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461.

Shen, Z.; and Kang, Z. 2025. When heterophily meets heterogeneous graphs: Latent graphs guided unsupervised representation learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Shi, C.; Hu, B.; Zhao, W. X.; and Yu, P. S. 2018. Heterogeneous information network embedding for recommendation. *IEEE transactions on knowledge and data engineering*, 31(2): 357–370.

Simonovsky, M.; and Komodakis, N. 2018. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, 412–422. Springer.

Tang, H.; Shao, L.; Sebe, N.; and Van Gool, L. 2024. Graph transformer GANs with graph masked modeling for architectural layout generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6): 4298–4313.

Tian, Y.; Dong, K.; Zhang, C.; Zhang, C.; and Chawla, N. V. 2023. Heterogeneous graph masked autoencoders. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 9997–10005.

Veli, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; Hjelm, D.; et al. 2019. Deep graph infomax. In *International Conference on Learning Representations*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wan, H.; Zhang, Y.; Zhang, J.; and Tang, J. 2019. Aminer: Search and mining of academic social networks. *Data Intelligence*, 1(1): 58–76.

Wang, H.; Wang, J.; Wang, J.; Zhao, M.; Zhang, W.; Zhang, F.; Xie, X.; and Guo, M. 2018. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019a. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019b. Heterogeneous graph attention network. In *The world wide web conference*, 2022–2032.

Wang, X.; Liu, N.; Han, H.; and Shi, C. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 1726–1736.

Wang, Y.; Sang, L.; Zhang, Y.; Zhang, Y.; and Wu, X. 2025. Generative-contrastive heterogeneous graph neural network. *IEEE Transactions on Big Data*.

Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, B.; and Bo, L. 2020. Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2397–2406.

Xia, R.; Cheng, Y.; Tang, Y.; Liu, X.; Liu, X.; Wang, L.; and Jiang, P. 2025. S-diff: An anisotropic diffusion model for collaborative filtering in spectral domain. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, 70–78.

Xiong, F.; Zhang, T.; Pan, S.; Luo, G.; and Wang, L. 2025. Robust Graph Based Social Recommendation Through Contrastive Multi-View Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12890–12898.

Yang, R.; Yang, Y.; Zhou, F.; and Sun, Q. 2023a. Directional diffusion models for graph representation learning. *Advances in Neural Information Processing Systems*, 36: 32720–32731.

Yang, T.; Hu, L.; Shi, C.; Ji, H.; Li, X.; and Nie, L. 2021. HGAT: Heterogeneous graph attention networks for semi-supervised short text classification. *ACM Transactions on Information Systems (TOIS)*, 39(3): 1–29.

Yang, X.; Yan, M.; Pan, S.; Ye, X.; and Fan, D. 2023b. Simple and efficient heterogeneous graph neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 10816–10824.

Yang, Y.; Guan, Z.; Wang, Z.; Zhao, W.; Xu, C.; Lu, W.; and Huang, J. 2022a. Self-supervised heterogeneous graph pre-training based on structural clustering. *Advances in Neural Information Processing Systems*, 35: 16962–16974.

Yang, Y.; Huang, C.; Xia, L.; Liang, Y.; Yu, Y.; and Li, C. 2022b. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2263–2274.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 974–983.

Yu, X.; Gu, Q.; Zhou, M.; and Han, J. 2012. Citation prediction in heterogeneous bibliographic networks. In *Proceedings of the 2012 SIAM international conference on data mining*, 1119–1130. SIAM.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 793–803.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1: 57–81.