Generating Graph-Like Logical Rules for Knowledge Graph Reasoning via Diffusion Models

Haoxiang Cheng*
Laboratory for Big Data and Decision, National University of Defense Technology
Changsha, China
hx_chenggfkd@nudt.edu.cn

Yunfei Wang*
National Key Laboratory of Information Systems Engineering, National University of Defense Technology
Changsha, China
wangyunfei@nudt.edu.cn

Chao Chen*
Laboratory for Big Data and Decision, National University of Defense Technology
Changsha, China
chenc1997@nudt.edu.cn

Kewei Cheng†
Microsoft Corporation
Redmond, USA
keweicheng@microsoft.com

Zhipeng Lin
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
linzhipeng13@nudt.edu.cn

Haoxuan Li
Peking University
Beijing, China
hxli@stu.pku.edu.cn

Changjun Fan‡
Laboratory for Big Data and Decision, National University of Defense Technology
Changsha, China
fanchangjun@nudt.edu.cn

Shixuan Liu‡
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
szftandy@hotmail.com

Abstract

Logical rules constitute a cornerstone of knowledge graph (KG) reasoning, valued for their interpretability and ability to model relational patterns. However, existing rule mining methods predominantly focus on simple chain-like rules and therefore neglect the richer relational information encoded in graph-like structures, such as cycles and branches. This limitation is further exacerbated by computational bottlenecks caused by the combinatorial explosion of the search space, which is especially challenging for graph-like rules. Meanwhile, generative approaches such as diffusion models, despite their success in other domains, cannot be directly applied to rule mining because their training objectives are not aligned with the goal of learning high-quality rules, and non-differentiable KG rule quality metrics cannot directly guide model optimization. To address these limitations, we propose GRiD, a framework that reformulates graph-like rule discovery as a discrete generative process conditioned on the target relation. GRiD employs a two-phase training strategy. First, supervised pre-training enables GRiD to capture structural priors from subgraphs sampled from the KG metagraph. Subsequently, reinforcement learning is applied to fine-tune GRiD through policy gradient optimization guided directly by non-differentiable rule-quality metrics. Experiments on six benchmark datasets show that GRiD achieves competitive performance on KG completion tasks. Ablation studies confirm the efficiency and robustness of GRiD and further show that graph-like rules complement chain-like rules in KG completion. Our code and datasets are available in https://github.com/Haoxiang-Cheng/GRiD.

CCS Concepts

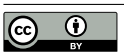
• **Computing methodologies** → *Semantic networks.*

Keywords

Knowledge Graph Reasoning; Logical Rules; Diffusion Models; Reinforcement Learning

ACM Reference Format:
Haoxiang Cheng, Yunfei Wang, Chao Chen, Kewei Cheng, Zhipeng Lin, Haoxuan Li, Changjun Fan, and Shixuan Liu. 2026. Generating Graph-Like Logical Rules for Knowledge Graph Reasoning via Diffusion Models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3817814

*These authors contributed equally to this research.
†Project Leader.
‡Corresponding authors.



KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817814

1 Introduction

Knowledge graphs (KGs) represent knowledge as factual triples (e_h, r, e_t) and serve as critical components in intelligent systems such as semantic search and question answering [33, 35, 46, 47].

However, KGs are inherently incomplete, necessitating reasoning methods to infer missing facts. Among various paradigms, rule-based reasoning offers interpretability by providing explicit logical rules that capture relational dependencies and enable transparent inference [15]. A logical rule typically expresses an implication of the form $\rho : \rho_h \leftarrow \rho_b$, where the rule head ρ_h can be inferred from a conjunction of body atoms ρ_b. In practice, rules are often induced from observed path instances, creating a direct link between schema semantics and instance-level evidence. For example, as illustrated in Fig. 1, the rule $\rho_1 : \mathit{BornIn}(x, z) \leftarrow \mathit{WorksAt}(x, y) \wedge \mathit{LocatedIn}(y, z)$ is induced from the instance path *BornIn(Turing, UK)* $\leftarrow$ *WorksAt(Turing, Cambridge)* $\wedge$ *LocatedIn(Cambridge, UK)*. Such instance-grounded induction yields transparent and verifiable reasoning paths, highlighting the utility of rule-based approaches.

Despite their transparency, traditional rule-based methods are largely confined to chain-like rules, which are sequences of relations connected by shared variables [7, 11, 23]. While chain-like rules are efficient to mine and apply, their linear structure limits their ability to express richer relational patterns, such as conjunctive or overlapping constraints. This restriction often leads to ambiguous predictions in KGs with complex schemas. For instance, applying the chain-like rule ρ_1 to infer *BornIn*(Hinton, ?) may yield multiple candidates, because the rule cannot distinguish among multiple candidate workplaces associated with the subject. This limitation arises from the inability of chain-like rules to jointly enforce multi-relational constraints, highlighting the need for more discriminative rule structures.

Graph-like rules, whose bodies form directed graphs rather than simple chains, can address this limitation by enforcing multiple relational constraints simultaneously. For instance, a graph-like rule such as $\rho_2 : \mathit{BornIn}(x, z) \leftarrow \mathit{WorksAt}(x, y) \wedge \mathit{GraduatedFrom}(x, y) \wedge \mathit{LocatedIn}(y, z)$ incorporates joint evidence through shared variables, thereby reducing ambiguity. However, extending rule learning from chain-like rules to general graph-like structures introduces a fundamental computational bottleneck. Even mining simple chain-like rules involves a combinatorial exponential search space, typically on the order of $O(|R|^L)$ for length L. Graph-like rules, which generalize chains by introducing non-linear dependencies, further expand this space to $O((L|R|)^L)$. This issue is inherent in search-based paradigms, particularly those extracting rules from instance paths, where traversing large KG instances incurs substantial computational overhead. Consequently, this challenge motivates a shift from iterative search to more efficient exploration of the rule space, in line with recent studies that formulate graph-structured combinatorial optimization as learnable optimization procedures [28].

While search-based rule discovery is constrained by combinatorial complexity, generative approaches have recently emerged as an alternative. Diffusion models, in particular, have shown effectiveness in generating structured data while maintaining robustness and diversity. Unlike Generative Adversarial Networks and Variational Autoencoders, which may suffer from mode collapse or over-smoothing [2], diffusion models can provide more stable coverage of complex output spaces. Moreover, while LLMs remain vulnerable to logical inconsistency, hallucination, and symbolic-translation errors in complex reasoning tasks [5, 6, 18, 44], diffusion models can incorporate explicit structural constraints to preserve rule validity. Nevertheless, existing diffusion-based methods in the KG domain primarily focus on instance-level tasks such as fact completion rather than on discovering explicit schema-level logical patterns [20, 21, 48].

Adapting diffusion models from instance-level reconstruction to schema-level rule induction is nontrivial. A central challenge lies in the non-differentiable nature of rule quality metrics. Unlike standard diffusion training, which optimizes differentiable reconstruction losses, logical rules are typically assessed using discrete quality metrics such as confidence and coverage. These metrics cannot be directly back-propagated, creating a mismatch between the training objective and the semantic quality of generated rules. As a result, naively applying diffusion models may fail to align generation with semantically meaningful rules.

To address these challenges, we propose GRiD, a framework for generating **G**raph-like **R**ules w**i**th **D**iffusion models for KG reasoning. GRiD reformulates rule discovery as a conditional discrete diffusion process over the rule-body adjacency matrix. To overcome the non-differentiability of discrete rule metrics, GRiD incorporates Reinforcement Learning (RL) to fine-tune the diffusion process using rule-quality metrics as reward signals. However, directly applying RL to the combinatorial space of graph-like rules often leads to severe sample inefficiency. To mitigate this issue, GRiD integrates a Supervised Learning (SL) pre-training strategy on subgraphs sampled from the KG meta-graph, which provides the denoising network with graph structure priors. By combining SL for structural pattern recognition with RL for quality optimization, GRiD enables the efficient generation of high-quality graph-like rules.

The contributions of this work are summarized as follows:

- We propose GRiD, a framework that reformulates rule discovery as a conditional generative process using a diffusion model, shifting this task from discrete search to generative modeling.
- We introduce a two-stage training strategy that employs SL pre-training to learn structure priors and RL fine-tuning to optimize the diffusion model using non-differentiable rule-quality metrics.
- Experiments show the effectiveness of GRiD for the KGC task, while ablation studies reveal the complementary effect of graph-like rules and the correlation between their effectiveness and KG structural properties.

2 Related Work

2.1 Logical Rule Learning

Logical rule learning is one of the main paradigms for KG reasoning, alongside embedding-based and path-based methods [15]. Compared with embedding-based and path-based methods, rule-based reasoning provides higher interpretability by deriving conclusions through explicit logical formulas. Rule-based reasoning applies predefined logical axioms and inference rules to derive new factual knowledge from existing KGs. Owing to this interpretability, substantial research has focused on developing methods for efficiently discovering high-quality logical rules from KGs.

Logical rule learning has evolved through three methodological stages. Early approaches primarily relied on inductive logic programming and association rule mining. Foundational systems such as FOIL [30] and AMIE [11] generate Horn clauses by generalizing

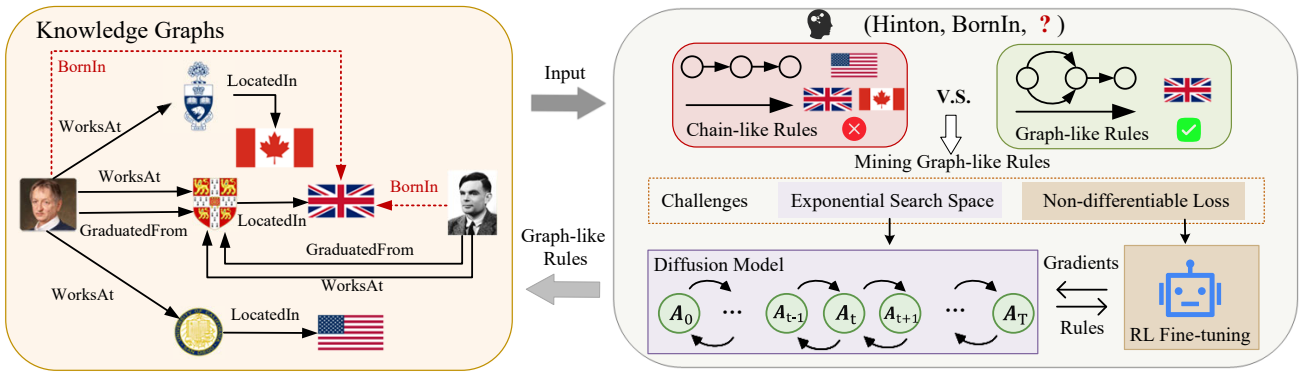


Figure 1: While existing rule-based methods predominantly rely on chain-like structures, such structures are insufficient for capturing complex interconnected patterns in KGs. Graph-like rules provide a more expressive alternative but are hindered by the exponential growth of the search space. Therefore, we propose a diffusion-based method for efficient rule generation. However, the standard supervised reconstruction objective for these models is misaligned with the goal of discovering semantically high-quality rules. We therefore introduce RL fine-tuning to directly optimize the generation policy using rewards based on structural quality, enabling the efficient generation of high-quality graph-like rules.

from specific instances, while other approaches are tailored to large graphs [10]. Subsequent research explored path-based methods, including AnyBURL [23], which samples and evaluates relational paths from KGs to construct logical rules. Related meta-path learning methods further address schema-complex heterogeneous information networks by learning schema-level path patterns without exhaustive path-instance enumeration [19]. While these methods offer interpretability, their dependence on paths or schema-level sequences limits their ability to capture non-linear graph-like rule structures.

Neural-symbolic integration paradigms were introduced to address the inefficiencies of discrete search. Early work in this direction, such as Neural Theorem Provers (NTPs) [31], introduced differentiable reasoning mechanisms. To reduce the computational demands of these approaches, subsequent research introduced optimizations including adaptive path selection [24] and dynamic rule subsetting [25]. Despite these improvements, scalability challenges continued to hinder practical deployment. Consequently, recent research has shifted toward neural-logic frameworks that reformulate rule learning as a continuous optimization problem. Neural-LP [43] introduced an attention-based controller for sequential rule construction, while hybrid architectures such as RNNLogic [29] jointly learn a rule generator and a reasoning predictor for the iterative refinement of rule embeddings. NCRL [7] employs a compositional learning strategy, combining rule components through neural representation learning to improve expressiveness.

Despite these advances, existing rule learning methods still focus mainly on chain-like structures, thereby overlooking graph-like structures that can model complex relational patterns in KGs. Moreover, generating graph-like rules is computationally challenging due to the exponential growth of the rule space. To bridge this gap, we leverage diffusion models to generate graph-like rules that capture complex relational patterns for KG reasoning.

2.2 Diffusion Models

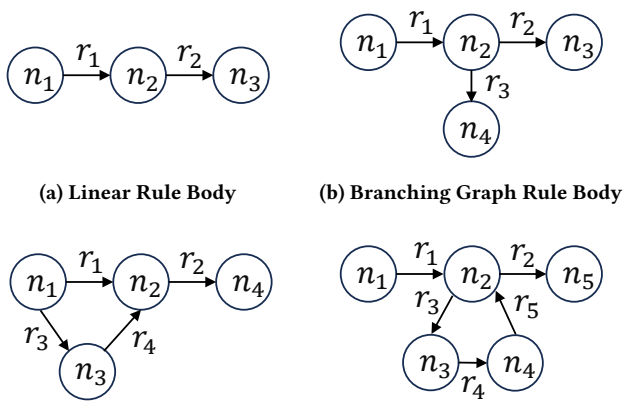
Diffusion models have demonstrated generative capabilities across a range of domains [45]. They have also been applied to graph-structured data generation, including molecular design, where they model complex constrained output spaces [13, 14, 40, 41]. This ability to model structured patterns under combinatorial constraints makes diffusion models suitable for KG reasoning tasks, which involve discovering valid relational paths within structured graphs.

Recent applications of diffusion models to KG reasoning can be grouped into two trajectories. The first strand operates at the instance level, reformulating KG reasoning as a conditional generation task in which denoising processes progressively recover target entities or facts [20**?**]. While this paradigm learns the distribution of valid facts and achieves competitive performance, it remains confined to instance-level prediction and does not explicitly capture schema-level logical patterns in KGs. The second trajectory applies diffusion processes in continuous latent spaces to learn fact distributions and model relational uncertainty [4]. This line of work improves representational flexibility and can model complex relational patterns. However, because these methods focus on refining entity representations rather than discovering schema structures, they provide limited interpretability.

While both paradigms demonstrate the adaptability of diffusion models to KG reasoning, neither directly generates explicit rules beyond the instance level, and therefore neither explicitly captures schema-level logical patterns. However, directly applying diffusion models to rule generation is challenging because standard reconstruction losses are not aligned with discrete rule-quality metrics. To address this issue, we employ RL to optimize the diffusion model using rule-quality metrics as reward signals, enabling the diffusion process to be applied to graph-like rule generation.

3 Preliminaries

This section formalizes the key concepts used in this work. We begin by reviewing the basic structure of KGs and logical rules. We



(a) Linear Rule Body **(b) Branching Graph Rule Body**

(c) Edge-cycle Graph Rule Body **(d) Node-cycle Graph Rule Body**

Figure 2: Four Basic Types of Graph-like Rule Bodies.

then introduce the concept of graph-like rules. Finally, we present four quantitative metrics for evaluating rule quality.

Definition 1 (Knowledge Graph, KG). A KG $\mathcal{G}$ is formally defined as a relation-labeled directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$, where $\mathcal{E}$ denotes the entity set, $\mathcal{R}$ denotes the relation set, and $\mathcal{F}$ denotes the fact set. Specifically, $\mathcal{F} = \{(e_h, r, e_t) | e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$ is the set of factual triples.

Definition 2 (Logical Rule). A logical rule ρ represents a logical implication between a head predicate and a body composed of conjunctive predicates. It has the following general form:

$$\rho : \rho_h \leftarrow \rho_b, \tag{1}$$

where ρ_h denotes the rule head, typically formulated as a single atom such as $r_h(e_i, e_j)$, and ρ_b denotes the rule body, which is formed by a conjunction of atoms. Semantically, the rule asserts that if the body ρ_b is true, then the head ρ_h is also true.

Definition 3 (Graph-like Rule). A graph-like rule is a logical rule whose body ρ_b forms a directed graph. As illustrated in Fig. 2, we categorize such rules into four basic types based on the connectivity pattern of the rule body: linear rule (Fig. 2a), which represents sequential dependencies; branching structure (Fig. 2b), which captures conjunctive paths; edge-cycling structure (Fig. 2c), which represents relation-centric cycles; and node-cycling structure (Fig. 2d), which models node-centric cycles. These basic structures can be combined to form more complex hybrid rule patterns.

Definition 4 (Support). The support of a logical rule ρ, denoted as $Supp_\rho^{\mathcal{G}}$, is the number of distinct entity pairs (e_i, e_j) in $\mathcal{G}$ for which both ρ_b and ρ_h hold. It is defined as follows:

$$Supp_\rho^{\mathcal{G}} := \#\{(e_i, e_j) \in \mathcal{G} : \mathbb{I}_{\rho_b}(e_i, e_j) \wedge \rho_h(e_i, e_j)\}. \tag{2}$$

Definition 5 (Coverage). The coverage of a rule ρ quantifies its prevalence among facts of the head relation. It is defined as the proportion of entity pairs connected by ρ_h that also satisfy ρ_b. It is defined as follows:

$$Cov_\rho^{\mathcal{G}} := \frac{Supp_\rho^{\mathcal{G}}}{\#\{(e_i, e_j) \in \mathcal{G} : \rho_h(e_i, e_j)\}}. \tag{3}$$

Definition 6 (Confidence). The confidence of a rule ρ measures the proportion of body groundings that also support the rule head. It is defined as follows:

$$Conf_\rho^{\mathcal{G}} := \frac{Supp_\rho^{\mathcal{G}}}{\#\{(e_i, e_j) \in \mathcal{G} : \mathbb{I}_{\rho_b}(e_i, e_j)\}}. \tag{4}$$

Definition 7 (PCA Confidence). PCA Confidence quantifies the precision of a rule ρ under the Partial Completeness Assumption (PCA) [11]. It restricts the denominator to body groundings whose subject entity has at least one observed fact under the head relation, and is defined as follows:

$$PCA\text{-}Conf_\rho^{\mathcal{G}} := \frac{Supp_\rho^{\mathcal{G}}}{\#\{(e_i, e_j') \in \mathcal{G} : \mathbb{I}_{\rho_b}(e_i, e_j') \wedge \rho_h(e_i, e_j')\}}. \tag{5}$$

4 Methods

This section presents GRiD, a framework that formulates graph-like rule discovery as a conditional discrete diffusion process over rule-body adjacency matrices. By leveraging a diffusion model, GRiD learns a distribution over KG rules conditioned on the target relation, thereby reducing dependence on explicit combinatorial search. GRiD operates in two stages: it is first pre-trained via SL to capture structural priors from subgraphs sampled from KG meta-graph and then fine-tuned with RL to optimize non-differentiable rule-quality metrics. An overview of the pipeline is provided in Fig. 3. The following sections detail the discrete diffusion framework, the two-stage training process, and the inference pipeline.

4.1 Discrete Diffusion Framework

GRiD adopts a discrete diffusion paradigm to model the symbolic structure of graph-like rules. While continuous latent representations may introduce ambiguity when reconstructing such structures, operating in the discrete space preserves their structural constraints. Accordingly, GRiD reformulates rule discovery as a conditional discrete diffusion process. The following subsections detail this framework, beginning with the discrete state formulation and followed by formal descriptions of the forward and reverse diffusion processes.

4.1.1 Discrete State Formulation. The process is conditioned on the rule head ρ_h, which provides the semantic target, while the rule body ρ_b is modeled as the adjacency matrix to be generated.

Discrete Matrix Representation. The rule body of graph-like rules is a directed graph, as defined in Def. 3, and can therefore be represented as a matrix. Formally, the rule body adjacency matrix $\mathbf{A}$ is defined as:

$$\mathbf{A} \in \{0, 1, \ldots, |\mathcal{R}|, \emptyset\}^{S \times S}. \tag{6}$$

Here, S denotes the maximum number of allowed nodes and $\mathcal{R}$ denotes the relation set of the KG. Each entry $\mathbf{A}[i, j] \in \{0, 1, \ldots, |\mathcal{R}|, \emptyset\}$ specifies the relation type from node i to node j, with a value of 0 indicating no relation and $\emptyset$ indicating padding. To accommodate variable-length rules, unused rows and columns are padded with a null token $\emptyset$, enabling fixed-dimensional processing for batch training.

Conditional Formulation. Generation is conditioned by two factors: the target head relation r_h and the diffusion timestep t. The

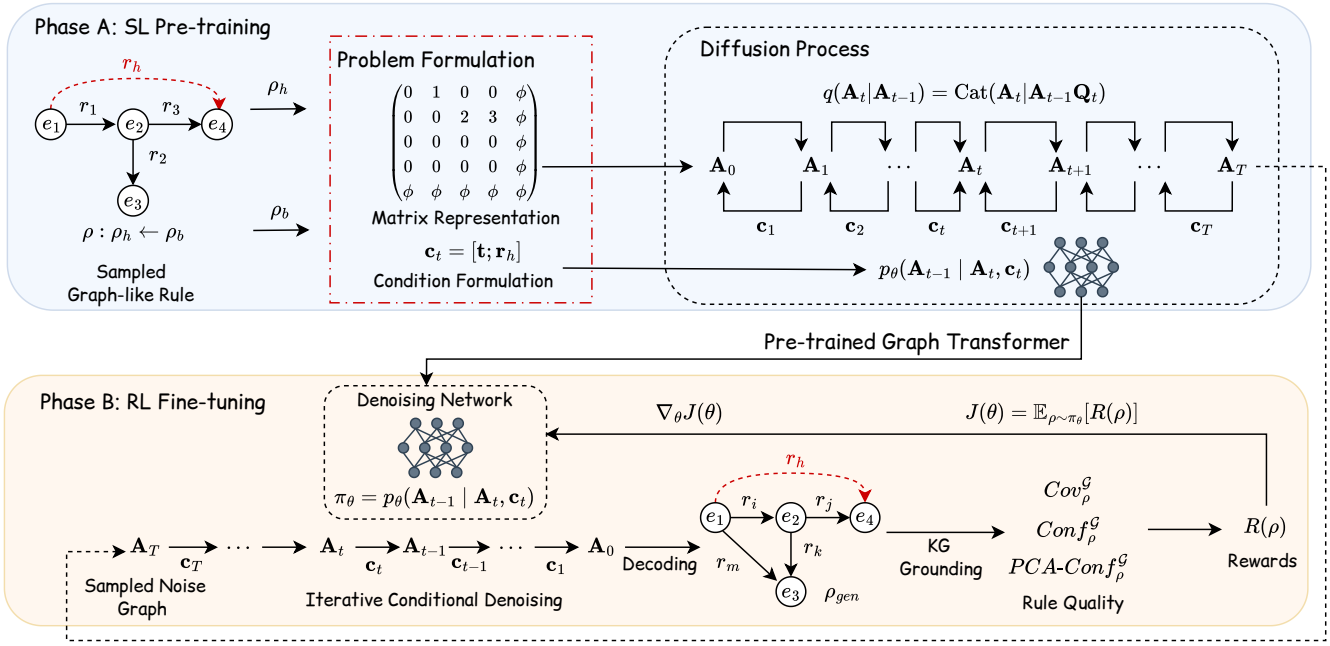


Figure 3: The GRiD framework. GRiD formulates graph-like rule discovery as a conditional discrete diffusion process over the rule-body adjacency matrices. Phase A (SL pre-training) learns graph-structure priors by corrupting subgraphs sampled from the KG meta-graph with categorical noise and training a Graph Transformer to reverse the diffusion process conditioned on the target relation. Phase B (RL fine-tuning) treats the denoising process as a stochastic policy, in which a rule is generated from a randomly initialized noisy graph through reverse diffusion, and the RL reward is computed from rule-quality metrics based on grounded rule instances in the KG. After RL fine-tuning, GRiD performs inference by reverse diffusion from sampled noise to generate top-ranked graph-like rules.

relation r_h is embedded into a static vector $\mathbf{r}_h \in \mathbb{R}^d$, while t is encoded via sinusoidal embeddings into a time-varying vector $\mathbf{t} \in \mathbb{R}^d$. These embeddings are concatenated to form a unified conditioning vector $\mathbf{c}_t$:

$$\mathbf{c}_t = [\mathbf{t}\,;\mathbf{r}_h] \in \mathbb{R}^{2d}. \tag{7}$$

The learning objective is to approximate the conditional distribution $p_\theta(\mathbf{A} \mid \mathbf{c}_t)$, so that the generated rule body is conditioned on r_h while respecting structural constraints at each diffusion step.

4.1.2 Forward Diffusion Process. The forward diffusion process progressively corrupts a clean rule-body adjacency matrix $\mathbf{A_0}$ into noise over T timesteps. Following the D3PM framework [1], the forward process is specified by a sequence of categorical transition matrices $\{\mathbf{Q}_t\}_{t=1}^T$. At each timestep t, the diffusion process factorizes across all edge positions. Each entry of $\mathbf{A}_t$ is represented as a K-dimensional one-hot categorical variable and evolves independently according to a categorical Markov transition:

$$q(\mathbf{A}_t \mid \mathbf{A}_{t-1}) = Cat(\mathbf{A}_t \mid \mathbf{A}_{t-1}\mathbf{Q}_t). \tag{8}$$

Here, $Cat(\mathbf{A}_t \mid \mathbf{P})$ denotes a collection of independent categorical distributions over all edge entries, where each probability vector is given by the corresponding row of $\mathbf{P}$, and $\mathbf{Q}_t \in \mathbb{R}^{K\times K}$ is a row-stochastic transition matrix over the $K = |\mathcal{R}| + 1$ edge types, including the no edge category. In this work, we adopt a uniform corruption scheme defined as follows:

$$\mathbf{Q}_t = (1-\beta_t)\mathbf{I} + \beta_t \frac{\mathbf{1}\mathbf{1}^\top}{K}, \tag{9}$$

where $\beta_t \in [0,1]$ represents the noise schedule. Here, $\mathbf{1} \in \mathbb{R}^K$ denotes an all-ones column vector, and $\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{K\times K}$ defines a uniform categorical transition. Under this construction, each edge is left unchanged with probability $1-\beta_t$ and is resampled uniformly from all K edge types with probability β_t. Using the Markov property, the marginal distribution at timestep t admits a closed form as follows:

$$q(\mathbf{A}_t \mid \mathbf{A}_0) = \mathrm{Cat}\Big(\mathbf{A}_t \;\mid\; \mathbf{A}_0\mathbf{Q}^{(t)}\Big), \quad \mathbf{Q}^{(t)} = \prod_{i=1}^{t}\mathbf{Q}_i, \tag{10}$$

which converges to a uniform categorical distribution as t increases, thereby removing information about the original rule structure.

4.1.3 Reverse Diffusion Process. This process iteratively denoises a noisy rule structure to reconstruct a clean rule-body adjacency matrix $\mathbf{A}_0$ from an initial state $\mathbf{A}_T$. The reverse transition $p_\theta(\mathbf{A}_{t-1} \mid \mathbf{A}_t, \mathbf{c}_t)$ is parameterized by a Graph Transformer f_θ, which takes the noisy adjacency matrix $\mathbf{A}_t$ and the conditioning vector $\mathbf{c}_t$ (Eq. 7) as input and outputs categorical logits for all edge types.

To condition generation on the current noise level and target relation, $\mathbf{c}_t$ is injected into each network layer via Feature-wise Linear

Modulation (FiLM) [27]. For the l-th layer, the node representation $\mathbf{H}^{(l)}$ is modulated as follows:

$$\mathbf{H}^{(l)} = \boldsymbol{\gamma}(\mathbf{c}_t) \odot \text{LayerNorm}(\mathbf{H}^{(l)}) + \boldsymbol{\beta}(\mathbf{c}_t), \tag{11}$$

where $\odot$ denotes element-wise multiplication, $\boldsymbol{\gamma}(\cdot)$ and $\boldsymbol{\beta}(\cdot)$ are learned affine projections of $\mathbf{c}_t$. This modulation enables the message-passing dynamics to adapt to both the timestep t and the target relation r_h.

Finally, the modulated representations are projected to produce a probability distribution over the K edge categories for each edge:

$$p_\theta(\mathbf{A}_{t-1} \mid \mathbf{A}_t, \mathbf{c}_t) \propto Cat(\mathbf{A}_{t-1} \mid f_\theta(\mathbf{A}_t, \mathbf{c}_t)). \tag{12}$$

By iteratively sampling from this distribution, GRiD generates a rule body conditioned on the target relation while preserving structural constraints.

4.2 Phase A: SL Pre-training

To obtain a structurally informed initialization, we pre-train GRiD to reconstruct subgraphs sampled from the KG meta-graph. The subgraph training set is constructed through random-walk sampling and topological augmentation on the KG, illustrated in Appendix G.

Training Objective. The objective is to train the denoising network f_θ to recover the clean adjacency matrix $\mathbf{A}_0$ from a corrupted state. Specifically, given a noisy input $\mathbf{A}_t$ and the conditioning vector $\mathbf{c}_t$, we minimize a weighted cross-entropy loss between the predicted edge-type distribution and the clean adjacency matrix:

$$\mathcal{L}_{recon} = -\frac{\sum_{i,j} M_{ij} \sum_{k=0}^{K-1} \mathbb{I}(\mathbf{A}_0[i,j]=k) \log p_\theta(\mathbf{A}_0[i,j]=k \mid \mathbf{A}_t, \mathbf{c}_t)}{\sum_{i,j} M_{ij}}. \tag{13}$$

Here, $\mathbf{M} \in \{0,1\}^{S\times S}$ is a binary mask, where $M_{ij}=0$ for padding positions and $M_{ij}=1$ otherwise. This ensures that the optimization focuses only on valid structural entries.

In addition to the reconstruction loss, we use the diffusion model to predict a scalar score $\hat{v}_{\rho_i}$ for each rule. The overall supervised pre-training objective is therefore defined as

$$\mathcal{L}_{SL} = \mathcal{L}_{recon} + \lambda_Q \frac{1}{B} \sum_{i=1}^{B} \left(\hat{v}_{\rho_i} - v^*_{\rho_i}\right)^2, \tag{14}$$

where $\lambda_Q = 0.1$ is a fixed hyperparameter that balances structural reconstruction and semantic regularization. The target quality score $v^*_{\rho_i}$ is defined as the min-max normalized sum of coverage and confidence ($Cov^{\mathcal{G}}_{\rho_i} + Conf^{\mathcal{G}}_{\rho_i}$) over the training set, thereby accounting for both rule recall and precision. This auxiliary objective aligns the learned representations with rule-quality signals and provides a warm start for subsequent RL fine-tuning.

Training Pipeline. Each supervised pre-training iteration corresponds to a single diffusion step conditioned on the target relation. Concretely, we first sample a batch of rule structures $\{(\mathbf{A}^i_0, r_h)\}_{i=1}^B$ from the training set, where B denotes the batch size. A diffusion timestep t is then drawn uniformly from $\{1,\dots,T\}$, and the corresponding noisy structure $\mathbf{A}_t$ is obtained through the forward diffusion process. Conditioned on the vector $\mathbf{c}_t$, GRiD predicts both the edge-type distributions and an auxiliary quality score for the corrupted structure. The supervised loss $\mathcal{L}_{SL}$ is then computed and back-propagated to update the model parameters θ.

4.3 Phase B: RL Fine-tuning

While SL pre-training establishes the structural prior from KG subgraphs, it does not directly optimize rule-quality metrics such as coverage and confidence. Inspired by a recent study [17], the pre-trained model is fine-tuned using RL to address this mismatch, which frames the sequential denoising process as a policy optimization problem.

Policy and Reward Formulation. We formulate the denoising process as a Markov Decision Process, where the denoising Graph Transformer acts as a stochastic policy $\pi_\theta(\mathbf{A}_{t-1} \mid \mathbf{A}_t, \mathbf{c}_t)$. The state corresponds to the noisy rule-body matrix $\mathbf{A}_t$, and the action is the sampled denoised matrix $\mathbf{A}_{t-1}$. Since intermediate noisy states lack direct rule-quality labels, the reward $R(\rho)$ is assigned only upon generating the complete rule ρ. To quantify rule quality, we define $R(\rho)$ as the geometric mean of coverage (Eq. 3), confidence (Eq. 4), and PCA confidence (Eq. 5):

$$R(\rho) = \left(Cov^{\mathcal{G}}_\rho \cdot Conf^{\mathcal{G}}_\rho \cdot PCA\text{-}Conf^{\mathcal{G}}_\rho\right)^{1/3}. \tag{15}$$

This composite metric balances rule generality and predictive precision, providing a reward signal for optimization.

Policy Optimization. Since the intermediate states do not have direct rule-quality labels, we formulate the objective as maximizing the expected episode-level reward $J(\theta) = \mathbb{E}_{\rho\sim\pi_\theta}[R(\rho)]$ using the REINFORCE algorithm. To reduce variance, we incorporate a baseline b, updated by an exponential moving average, to compute the advantage $\hat{A} = R(\rho) - b$. The optimization objective $\mathcal{L}_{RL}$ combines the policy gradient with entropy regularization to discourage premature convergence to deterministic transitions and encourage exploration:

$$\mathcal{L}_{RL}(\theta) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{t=1}^{T}\left(\hat{A}_i \log \pi_\theta(\mathbf{A}^{(i)}_{t-1} \mid \mathbf{A}^{(i)}_t, \mathbf{c}^{(i)}_t) + \lambda_H \mathcal{H}(\pi_\theta(\cdot \mid \mathbf{A}^{(i)}_t, \mathbf{c}^{(i)}_t))\right). \tag{16}$$

Here, $\hat{A}_i$ acts as a trajectory-level weight for episode i, reinforcing steps that lead to higher-quality rules, while the entropy term $\mathcal{H}$ encourages exploration at each timestep.

4.4 Inference Pipeline

After training, the diffusion model is used as a conditional generator of graph-like rule bodies for a given target relation r_h. Inference proceeds via iterative denoising followed by structural parsing and filtering. Specifically, we initialize the process from a noisy adjacency matrix $\mathbf{A}_T$, where each entry is sampled from the K edge categories. From $t=T$ to $t=1$, the model iteratively samples

$$\mathbf{A}_{t-1} \sim p_\theta(\mathbf{A}_{t-1} \mid \mathbf{A}_t, \mathbf{c}_t), \tag{17}$$

where the conditioning vector $\mathbf{c}_t = [\mathbf{t}; \mathbf{r}_h]$ encodes the timestep and target relation. This denoising process yields a discrete adjacency matrix $\mathbf{A}_0$, which is then parsed into a graph-like rule body.

To form a rule, the node with zero in-degree is identified as the head variable e_h. The node with zero out-degree that is topologically farthest from e_h is designated as the tail variable e_t. The remaining edges compose the rule body that predicts $r_h(e_h, e_t)$. Finally, KG structural constraints are applied to filter invalid structures, retaining valid graph-like rules for downstream reasoning.

Table 1: KGC results of GRiD on six datasets. The best/second-best metrics are bolded/underlined. All reported results are averaged over five independent runs with different random seeds, detailed in Appendix E.

	Models	Family			Kinship			UMLS		
		MRR (%)	HIT@1 (%)	HIT@10 (%)	MRR (%)	HIT@1 (%)	HIT@10 (%)	MRR (%)	HIT@1 (%)	HIT@10 (%)
Embedding	TransE	45.21	22.34	87.45	31.23	0.93	83.22	69.86	52.14	89.19
	DistMult	54.17	36.00	88.45	35.23	19.22	74.23	39.04	25.61	66.15
	ComplEx	81.11	72.12	93.63	42.12	23.56	82.11	41.29	27.93	70.92
	RotatE	86.70	77.43	93.21	66.21	50.24	**94.13**	74.21	63.92	94.04
	SpeedE	90.28	82.72	98.47	65.32	53.35	89.55	70.35	58.62	89.65
GNN-based	CompGCN	75.35	58.83	88.83	40.30	26.94	75.83	62.33	50.85	72.76
	NBFNet	84.16	76.03	97.71	49.54	34.43	81.07	70.74	64.19	82.48
	A*Net	82.35	72.37	96.67	44.36	30.42	77.36	66.37	53.63	78.37
Rule-based	AnyBURL	93.99	88.46	94.90	65.93	52.38	91.45	68.83	67.40	72.54
	AMIE	87.74	80.47	98.83	62.99	48.83	90.06	64.42	60.67	74.21
	Neural-LP	88.12	79.28	98.53	30.17	16.60	59.34	48.38	33.28	77.47
	DRUM	89.03	82.76	99.06	33.02	18.11	67.76	55.18	35.37	85.15
	RNNLogic	86.74	79.40	95.18	64.78	49.70	92.26	75.20	62.40	92.04
	RLogic	88.08	80.81	97.27	57.70	43.60	87.08	71.74	55.32	93.55
	NCRL	91.33	85.24	**99.31**	64.32	49.39	92.56	75.53	64.44	93.15
	ChatRule	90.62	85.45	96.84	32.52	17.51	68.74	77.51	67.45	94.84
	GRiD	**95.25**	**91.54**	98.32	**67.18**	**54.36**	92.46	**83.48**	**74.89**	**96.48**

	Model	WN-18RR			FB15K-237			YAGO3-10		
		MRR (%)	HIT@1 (%)	HIT@10 (%)	MRR (%)	HIT@1 (%)	HIT@10 (%)	MRR (%)	HIT@1 (%)	HIT@10 (%)
Embedding	TransE	23.12	2.23	52.54	29.40	18.65	46.84	36.42	25.78	58.01
	DistMult	42.67	38.06	50.38	22.53	13.93	38.46	34.56	24.54	52.93
	ComplEx	44.11	41.35	51.15	24.49	15.26	41.67	33.34	24.23	53.86
	RotatE	47.17	42.92	55.74	32.40	22.24	52.48	49.41	40.76	**67.70**
	SpeedE	49.31	44.52	57.63	32.71	22.32	49.57	41.35	33.25	51.35
GNN-based	CompGCN	45.32	39.25	54.63	33.74	22.98	52.55	28.43	21.36	47.52
	NBFNet	**53.87**	**49.50**	62.93	39.22	28.79	57.39	36.83	27.91	55.42
	A*Net	51.18	42.82	**64.37**	39.88	30.89	58.13	36.16	27.36	55.93
Rule-based	AnyBURL	41.20	37.91	47.90	41.32	30.97	61.40	46.14	39.60	58.52
	AMIE	36.02	33.42	41.00	-	-	-	-	-	-
	Neural-LP	37.67	36.46	40.15	24.15	17.71	36.56	-	-	-
	DRUM	36.68	38.41	41.30	22.89	17.23	35.75	-	-	-
	RNNLogic	46.28	41.40	53.32	28.59	20.20	44.62	-	-	
	RLogic	44.23	40.14	50.01	31.10	19.78	50.02	36.23	25.23	50.25
	NCRL	46.68	44.64	53.22	29.76	20.24	47.46	38.33	27.87	53.14
	ChatRule	33.51	30.12	40.43	30.91	22.32	49.88	44.92	35.41	62.70
	GRiD	45.10	40.22	54.39	**42.29**	**32.05**	**62.13**	**54.91**	**49.78**	62.97

5 Experiment

5.1 Experimental settings

Tasks. We evaluate the performance and efficiency of GRiD on KGC tasks. The objective is to predict a missing target entity e_t for a given query $(e_h, r_q, ?)$. KGC tasks assess the capability of GRiD to extract high-quality logical rules for KG reasoning. The full KGC pipeline is detailed in Appendix F.

Datasets. Our evaluation spans six benchmark datasets representing diverse scales and domains: two large-scale KGs, YAGO3-10 [34] and FB15K-237 [37], alongside four domain-specific datasets: Family [12], Kinship [16], UMLS [16], and WN-18RR [9]. Detailed statistics are provided in Appendix A.

Dataset Preparation. Each dataset is randomly split into training and test sets with a ratio of 9:1. To avoid information leakage, all test triples are strictly excluded from the training data.

Baselines. We compare GRiD against 16 representative baselines spanning three paradigms under identical experimental settings: five embedding approaches, including TransE [3], DistMult [42], ComplEx [38], RotatE [36], and SpeedE [26]; three GNN-based methods, including CompGCN [39], NBFNet [50], and A*Net [49]; and eight rule-based systems, including AnyBURL [23], AMIE [11], Neural-LP [43], DRUM [32], RNNLogic [29], RLogic [8], NCRL [7], and ChatRule [22]. Detailed descriptions of these baselines are provided in Appendix B.

Metrics. Following standard KGC evaluation protocols, we adopt filtered ranking, where triples $(e_h, r_q, e^\star)$ with $e^\star \neq e_t$ are removed from the candidate set if they exist in the KG. We report Hit@1, Hit@10, and Mean Reciprocal Rank (MRR).

Rule Application. For each relation r, both chain-like and graph-like rules are used to compute the final scoring matrix, $\mathbf{S}_r = (1-\alpha)\mathbf{S}_r^{\text{chain}} + \alpha\mathbf{S}_r^{\text{graph}}$, where the fusion weight is $\alpha = 0.1$. The detailed scoring and aggregation procedure is described in Appendix F.

5.2 Knowledge Graph Completion Results

Overall Results. As summarized in Table 1, GRiD achieves consistently strong KGC performance across six benchmarks. It outperforms most rule-based baselines in the majority of settings and remains competitive with representative embedding and GNN-based baselines, while producing explicit logical rules. On large-scale KGs such as FB15K-237 and YAGO3-10, several rule-based approaches encounter scalability limitations, whereas GRiD still achieves the best or second-best results.

Impact of KG Structure. We observe that graph-like rules are particularly beneficial on KGs where rich relational structure leads to ambiguous predictions under simple chain-based reasoning, such as FB15K-237 and YAGO3-10. In such settings, chain-like rules often yield ambiguous candidate rankings, while graph-like rules introduce additional conjunctive constraints that help disambiguate candidate entities. Conversely, on KGs with tightly constrained relational semantics (e.g., kinship or lexical hierarchies), chain-like rules already capture most reliable patterns, leaving less room for improvement from graph-like rules. A detailed quantitative analysis of KG structural properties and their relationship to performance is provided in Appendix C.

Case Study. To further illustrate the differences between the learned logical rules, representative examples of chain-like and graph-like rules are presented in Appendix D. Graph-like rules impose stricter conjunctive constraints by requiring the joint satisfaction of multiple relational conditions. For example, the rule $isCitizenOf(x, y) \leftarrow LivesIn(x, y) \wedge WorksAt(x, y)$ integrates two complementary relational signals through the shared variables (x, y). In contrast, chain-like rules are characterized by a single linear relational dependency.

5.3 Ablation Studies

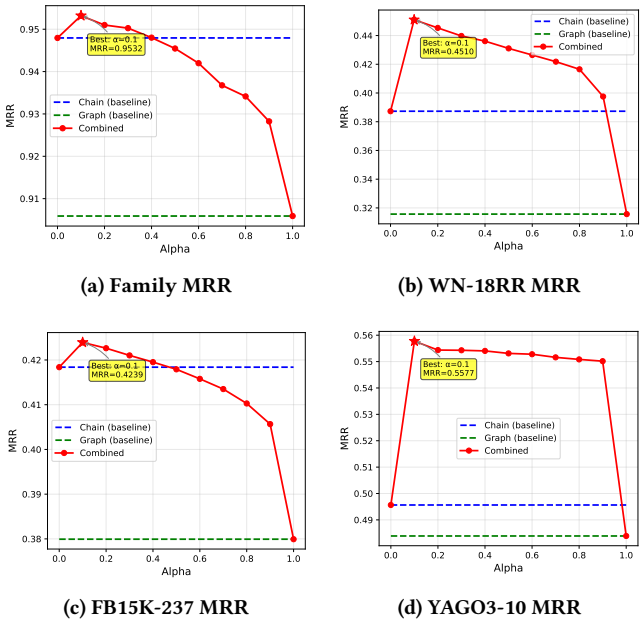
We conduct ablation studies to assess several design choices of GRiD, focusing on four key questions: **(Q1) Effectiveness of Graph-like Rules:** Do graph-like rules provide complementary inferential evidence beyond chain-like rules? **(Q2) Sensitivity to Fusion Weight:** Under which fusion weights do graph-like rules provide the largest performance gains? **(Q3) Necessity of RL:** Is RL fine-tuning essential for rule discovery? **(Q4) Size of Graph-like Rules:** Is a small graph size sufficient for complex KGs?

(Q1) Effectiveness of Graph-like Rules. To address Q1, we investigate whether graph-like rules provide complementary inferential evidence beyond chain-like rules. We compare three configurations: *Chain-only*, *Graph-only*, and their *Combination*, while maintaining a constant total number of rules for a controlled comparison. As shown in Table 2, the combined configuration consistently outperforms either individual type, indicating that graph-like rules capture relational patterns distinct from those encoded by chain-like rules. Graph-like rules in isolation achieve lower KGC performance because their stricter constraints reduce coverage in sparse KGs, which is consistent with the sparsity of KG observations.

(Q2) Sensitivity to Fusion Weight. To address Q2, we vary the fusion weight α at the scoring stage while keeping the generated chain-like and graph-like rule sets fixed. Here, $\alpha = 0$ corresponds to chain-only reasoning, and $\alpha = 1$ corresponds to graph-only reasoning. As shown in Fig. 4, adding graph-like rules improves the chain-only baseline, with the best performance typically obtained when α is around 0.1–0.2. Larger α values degrade performance because graph-like rules enforce additional joins, which improve precision but reduce the number of supported groundings. These results show that the fusion score balances the coverage of chain-like rules with the stricter constraints of graph-like rules.

Table 2: KGC Results of Different Rule Types

Rule-Type	Family			WN-18RR		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
Chain-only	94.79	91.21	99.01	38.73	35.12	45.56
Graph-only	90.59	84.72	97.60	31.57	27.18	40.26
Combined	**95.32**	**91.66**	**99.65**	**45.10**	**40.22**	**54.39**
Rule-Type	FB15K-237			YAGO3-10		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
Chain-only	41.84	31.82	61.08	49.56	44.96	56.78
Graph-only	37.99	28.64	55.60	48.39	43.84	55.44
Combined	**42.29**	**32.05**	**62.13**	**55.77**	**49.78**	**62.97**



(a) Family MRR (b) WN-18RR MRR

(c) FB15K-237 MRR (d) YAGO3-10 MRR

Figure 4: Ablation study on the fusion weight α.

(Q3) Necessity of RL. To address Q3, we compare GRiD trained with only supervised pre-training against two RL-fine-tuned variants on YAGO3-10: one using a geometric-mean reward and the other using the sum of rule-quality metrics. As shown in Fig. 5, RL fine-tuning yields consistent improvements of approximately 10% across all metrics, indicating that direct optimization toward rule quality helps align generation with reasoning objectives. Moreover, the geometric-mean reward outperforms metric summation, suggesting that balanced multi-criteria optimization provides a more effective training signal in this setting.

(Q4) Size of Graph-like Rules. To address Q4, we conduct KGC experiments with different graph-size limits S. As shown in Table 3, $S = 6$ consistently achieves the best MRR across all datasets. This

Table 3: KGC MRR under different graph-size limits S.

S	WN-18RR	FB15K-237	YAGO3-10
5	0.420	0.381	0.518
6	**0.462**	**0.411**	**0.548**
7	0.452	0.397	0.519

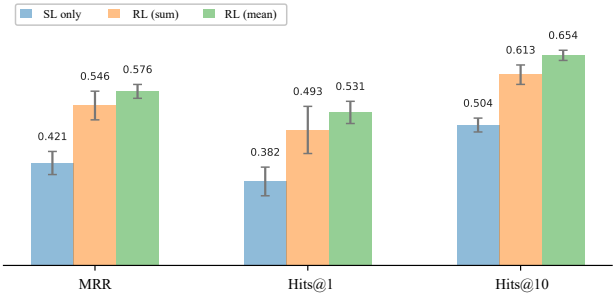


Figure 5: Effect of RL fine-tuning on YAGO3-10.

result is consistent with the compositional nature of logical rules: longer rules often provide limited additional utility because they can be expressed by composing shorter rules. Moreover, rules operate at the schema level and are therefore not related to instance-level neighbor density. Consequently, a small S can improve efficiency while preserving sufficient expressiveness for the rule patterns considered in this work.

5.4 Efficiency Analysis

We assess the efficiency of GRiD from both theoretical and empirical perspectives. In SL pre-training, the cost is dominated by the Graph Transformer, with per-epoch complexity $O(|\mathcal{D}_{pre}|S^2)$. In RL fine-tuning, complete denoising trajectories are sampled for reward computation, giving complexity $O(N_{RL}TS^2)$ for N_{RL} sampled rules and T diffusion steps. Since graph-like rule bodies are compact ($S \leq 6$), the quadratic dependence on S introduces limited overhead. Rule generation is performed offline for each relation; therefore, the online cost for each query is independent of the denoising trajectory length T and mainly comes from sparse-matrix lookup and aggregation over retained rule groundings. Thus, the denoising cost affects offline rule construction rather than KGC inference.

Empirically, as reported in Table 4, GRiD has moderate training cost and memory usage. On YAGO3-10, SL pre-training and RL fine-tuning require 1.34 s and 18.76 s per epoch, respectively. Peak GPU memory is 11.41 GB on FB15K-237 and 5.52 GB on YAGO3-10. Although sampling on large KGs is more time-consuming, it is an offline cost incurred once per relation. After rule generation, inference is performed by sparse-matrix grounding and aggregation, supporting rule application on large-scale KGs.

6 Conclusion

This paper introduces GRiD, a framework that reformulates rule discovery as a discrete diffusion process conditioned on the target relation. By shifting from iterative search to generative modeling, GRiD reduces the dependence on exhaustive search in the combinatorial rule space. By combining SL pre-training for structural priors with RL fine-tuning that directly optimizes rule quality metrics, GRiD generates graph-like rules for KG reasoning. Experiments show that the rules generated by GRiD support KGC across six benchmark datasets. Furthermore, ablation studies show the complementary effects of graph-like rules and chain-like rules. Future research will explore generating rules directly from underlying KG distributions to mitigate sampling bias and developing structural evaluation metrics that provide more precise guidance for the diffusion process.

Table 4: Mean computational cost over 5 independent runs. Time is reported in seconds (s) and peak memory in GB.

Dataset	Sampling (s)	SL/epoch (s)	RL/epoch (s)	Gen/100 (s)	Total Time (s)	Peak Mem (GB)
Family	4.23	0.74	1.35	1.10	58.91	1.81
Kinship	4.87	0.14	0.46	0.42	223.67	3.57
UMLS	10.11	1.67	2.89	3.15	125.59	6.06
WN-18RR	5.84	0.21	2.57	4.44	93.99	1.54
FB15K-237	3079.35	8.61	4.96	0.99	3699.88	11.41
YAGO3-10	4479.54	1.34	18.76	1.03	7279.68	5.52

7 Acknowledgements

This study was supported by the National Natural Science Foundation of China (Nos. 72571284, 72421002, 62273352), the Hunan Provincial Fund for Distinguished Young Scholars (No. 2025JJ20073), the Science and Technology Innovation Program of Hunan Province (No. 2023RC3009), the Innovation Research Foundation of National University of Defense Technology (No. JS24-05), the Major Science and Technology Projects in Changsha, China (No. kq2301008), the Beijing Natural Science Foundation (No. L257007), the Beijing Major Science and the Technology Project (No. Z251100008425006).

References

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 17981–17993.

[2] Serguei Barannikov, Ilya Trofimov, Grigorii Sotnikov, Ekaterina Trimbach, Alexander Korotin, Alexander Filippov, and Evgeny Burnaev. 2021. Manifold Topology Divergence: A Framework for Comparing Data Manifolds. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 7294–7305.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 26. 2787–2795.

[4] Yuxiang Cai, Qiao Liu, Yanglei Gan, Changlin Li, Xueyi Liu, Run Lin, Da Luo, and JiayeYang JiayeYang. 2024. Predicting the Unpredictable: Uncertainty-Aware Reasoning over Temporal Knowledge Graphs via Diffusion Process. In *Findings of the Association for Computational Linguistics: ACL 2024 (ACL Findings)*. Association for Computational Linguistics, 5766–5778.

[5] Zheng Chen, Chuan Zhou, Fengxiang Cheng, Yip Tin Po, Fenrong Liu, Yisen Wang, Jiajun Chai, Xiaohan Wang, Guojun Yin, Wei Lin, Bo Li, Haoxuan Li, and Zhouchen Lin. 2026. LogiConBench: Benchmarking Logical Consistencies of LLMs. In *The Fourteenth International Conference on Learning Representations*.

[6] Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert Van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering LLMs with logical reasoning: a comprehensive survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*. 10400–10408.

[7] Kewei Cheng, Nesreen K. Ahmed, and Yizhou Sun. 2023. Neural Compositional Rule Learning for Knowledge Graph Reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[8] Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. RLogic: Recursive Logical Rule Learning from Knowledge Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 179–189.

[9] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 32.

[10] Wenfei Fan, Xin Wang, Yinghui Wu, and Jingbo Xu. 2015. Association Rules with Graph Patterns. *Proceedings of the VLDB Endowment (PVLDB)* 8, 12 (2015), 1502–1513.

[11] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. ACM, 413–422.

[12] Geoffrey E. Hinton. 1986. Learning Distributed Representations of Concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, 1–12.

[13] Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. Conditional diffusion based on discrete graph structures for molecular graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4302–4311.

[14] Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. 2024. Learning Joint 2-D and 3-D Graph Diffusion Models for Complete Molecule Generation. *IEEE Transactions on Neural Networks and Learning Systems* 35, 9 (2024), 11857–11871.

[15] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2022), 494–514.

[16] Stanley Kok and Pedro Domingos. 2007. Statistical Predicate Invention. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*. ACM, 433–440.

[17] Zhiqiang Kou, Junyang Chen, Xin-Qiang Cai, Xiaobo Xia, Ming-Kun Xie, Dong-Dong Wu, Biao Liu, Yuheng Jia, Xin Geng, Masashi Sugiyama, et al. 2026. Positive-Unlabeled Reinforcement Learning Distillation for On-Premise Small Models. *arXiv preprint arXiv:2601.20687* (2026).

[18] Shixuan Liu, Haoxiang Cheng, Yunfei Wang, Yue He, Changjun Fan, and Zhong Liu. 2025. EvoPath: Evolutionary Meta-Path Discovery with Large Language Models for Complex Heterogeneous Information Networks. *Information Processing & Management* 62, 1 (2025), 103920.

[19] Shixuan Liu, Changjun Fan, Kewei Cheng, Yunfei Wang, Peng Cui, Yizhou Sun, and Zhong Liu. 2024. Inductive meta-path learning for schema-complex heterogeneous information networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[20] Xiao Long, Liansheng Zhuang, Aodi Li, Houqiang Li, and Shafei Wang. 2024. Fact Embedding through Diffusion Model for Knowledge Graph Completion. In *Proceedings of the ACM Web Conference (WWW)*. ACM, 2020–2029.

[21] Xiao Long, Liansheng Zhuang, Aodi Li, Jiuchang Wei, Houqiang Li, and Shafei Wang. 2024. KGDM: A Diffusion Model to Capture Multiple Relation Semantics for Knowledge Graph Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8850–8858.

[22] Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2025. ChatRule: Mining Logical Rules with Large Language Models for Knowledge Graph Reasoning. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Springer, 342–355.

[23] Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckenschmidt. 2024. Anytime bottom-up rule learning for large-scale knowledge graph completion: C. Meilicke et al. *The VLDB Journal* 33, 1 (2024), 131–161.

[24] Pasquale Minervini, Matko Bošnjak, Tim Rocktäschel, and Sebastian Riedel. 2018. Towards Neural Theorem Proving at Scale. In *Proceedings of the ICML Workshop on Neural Abstract Machines and Program Induction (NAMPI)*.

[25] Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020. Learning Reasoning Strategies in End-to-End Differentiable Proving. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, 6938–6949.

[26] Aleksandar Pavlović and Emanuel Sallinger. 2024. SpeedE: Euclidean Geometric Knowledge Graph Embedding Strikes Back. In *Findings of the Association for Computational Linguistics: NAACL 2024 (NAACL Findings)*. Association for Computational Linguistics, 89–98.

[27] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 32.

[28] Tianle Pu, Chao Chen, Li Zeng, Shixuan Liu, Rui Sun, and Changjun Fan. 2024. Solving Combinatorial Optimization Problem over Graph through QUBO Transformation and Deep Reinforcement Learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. IEEE, 390–399.

[29] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2021. RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[30] J. Ross Quinlan. 1990. Learning Logical Definitions from Relations. *Machine Learning* 5 (1990), 239–266.

[31] Tim Rocktäschel and Sebastian Riedel. 2017. End-to-End Differentiable Proving. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30.

[32] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: End-to-End Differentiable Rule Mining on Knowledge Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32.

[33] Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-Sequence Knowledge Graph Completion and Question Answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2814–2828.

[34] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*. ACM, 697–706.

[35] Yushi Sun, Kai Sun, Yifan Ethan Xu, Xiao Yang, Xin Luna Dong, Nan Tang, and Lei Chen. 2025. KERAG: Knowledge-Enhanced Retrieval-Augmented Generation for Advanced Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. 6194–6216.

[36] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[37] Kristina Toutanova and Danqi Chen. 2015. Observed versus Latent Features for Knowledge Base and Text Inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. Association for Computational Linguistics, 57–66.

[38] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. PMLR, 2071–2080.

[39] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-Based Multi-Relational Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[40] Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. 2023. DiGress: Discrete Denoising Diffusion for Graph Generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[41] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. 2023. De novo design of protein structure and function with RFdiffusion. *Nature* 620, 7976 (2023), 1089–1100.

[42] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[43] Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30. 2316–2325.

[44] Haocheng Yang, Fengxiang Cheng, Tianjun Yao, Mengyue Yang, Jiajun Chai, Xiaohan Wang, Guojun Yin, Wei Lin, Soummya Kar, Fenrong Liu, Haoxuan Li, and Yisen Wang. 2026. MAD-Logic: Multi-Agent Debate Enhances Symbolic Translation and Reasoning. In *The Fourteenth International Conference on Learning Representations*.

[45] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion Models: A Comprehensive Survey of Methods and Applications. *Comput. Surveys* 56, 4 (2023), 1–39.

[46] Hang Yin, Zihao Wang, Weizhi Fei, and Yangqiu Song. 2025. EFO$_k$-CQA: Towards Knowledge Graph Complex Query Answering beyond Set Operation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 5876–5887.

[47] Yu Zhang, Kehai Chen, Xuefeng Bai, Zhao Kang, Quanjiang Guo, and Min Zhang. 2024. Question-guided knowledge graph re-scoring and injection for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 8972–8985.

[48] Cai Zhou, Xiyuan Wang, and Muhan Zhang. 2024. Unifying Generation and Prediction on Graphs with Latent Graph Diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 37. 61963–61999.

[49] Zhaocheng Zhu, Xinyu Yuan, Michael Galkin, Louis-Pascal Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. 2023. A*Net: A Scalable Path-Based Reasoning Approach for Knowledge Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36. 59323–59336.

[50] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2021. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 29476–29490.

Appendices

Appendix A Dataset Description

- **Family** [12] captures relations among family members.
- **Kinship** [16] contains kinship relations among members of the Alyawarra tribe from Central Australia.
- **UMLS** [16] contains biomedical concepts as entities and treatment and diagnosis relations.
- **WN-18RR** [9] is a lexical dataset whose entities refer to word senses and whose relations define lexical connections.
- **FB15K-237** [37] is derived from Freebase, an online collection of structured data extracted from multiple sources.
- **YAGO3-10** [34] is a subset of YAGO, a large semantic knowledge base constructed from multiple sources.

Appendix B Baseline Description

Embedding Baselines.

- **TransE** [3]. TransE models relations as translations between entity embeddings, capturing one-hop relational patterns.
- **DistMult** [42]. DistMult uses diagonal bilinear scoring for entity-relation interactions.
- **ComplEx** [38]. ComplEx uses complex embeddings to capture symmetric and antisymmetric relations.
- **RotatE** [36]. RotatE models relations as rotations in complex space and captures inversion patterns.
- **SpeedE** [26]. SpeedE is a lightweight Euclidean geometric KGE model for efficient KGC.

GNN-based Baselines.

- **CompGCN** [39]. CompGCN jointly embeds nodes and relations via compositional GCN aggregation.
- **NBFNet** [50]. NBFNet learns Bellman-Ford-style path representations for link prediction through neural path aggregation over relation-specific message passing.
- **A*Net** [49]. A*Net extends NBFNet with A*-like heuristic search to prioritize informative paths and reduce unnecessary expansion during reasoning over large KGs.

Rule-based Baselines.

- **AnyBURL** [23]. AnyBURL mines rules via anytime bottom-up path sampling and confidence-based rule selection.
- **AMIE** [11]. AMIE mines Horn rules in large KBs under Partical Compeletness Assumption for scalable rule evaluation.
- **Neural-LP** [43]. Neural-LP learns logical rules through differentiable tensor operations.
- **DRUM** [32]. DRUM extends Neural-LP with RNNs to model longer rule chains and recurrent dependencies.
- **RNNLogic** [29]. RNNLogic uses an RNN-based generator to propose logical rules for KG reasoning.
- **RLogic** [8]. RLogic introduces recursive logical rule learning through predicate representation learning and recursive rule composition.
- **NCRL** [7]. NCRL is a neural compositional rule learning method that uses attention mechanisms for recursive reasoning over rule components.
- **ChatRule** [22]. ChatRule uses prompts to guide LLMs in generating logical rules with KG structural validation.

Appendix C Impact of KG Structural Properties

To examine when graph-like rules are useful, we analyze the structural indicators, including schema-level connectivity (BF_Schema), relation frequency imbalance (Gini), relation co-occurrence tendency (NPIM), and the many-to-many (N-to-N) ratio, in Table 7 and report Graph Better, the proportion of relations where graph-like rules outperform chain-like rules.

Graph-like rules provide limited gains on Family, Kinship, and WN-18RR for different reasons. Family and Kinship contain semantically well-defined kinship relations, where chain-like rules already capture the main reasoning patterns. WN-18RR is instead limited by weak relation co-occurrence, with many relations being mutually exclusive rather than jointly informative (NPIM = -0.07). Therefore, graph constraints provide limited additional evidence on these datasets.

In contrast, UMLS, FB15K-237, and YAGO3-10 show higher Graph Better values. These datasets have more skewed relation distributions and richer relational ambiguity, where chain-like rules may match multiple candidates. Graph-like rules are therefore more useful because their conjunctive constraints help reduce ambiguity.

Overall, graph-like rules are most beneficial when additional structural constraints can disambiguate chain-based reasoning, but their advantage is limited when relation semantics are already regular or weakly co occurring.

Appendix D Rule Comparison

Representative chain-like and graph-like rules generated by GRiD on YAGO3-10 are presented in Table 5. Here, we analyze graph-like rules according to the four rule-body types in Fig. 2.

Linear Rule Body (Fig. 2a). This type represents a sequential variable chain. The rule $isCitizenOf(x, y) \leftarrow hasAcademicAdvisor(x, z_1) \wedge worksAt(z_1, z_2) \wedge isLocatedIn(z_2, y)$ demonstrates this structure, where inference follows a single path from x to y through the workplace of the advisor and its location. This rule captures citizenship through the institutional affiliation of the advisor.

Branching Rule Body (Fig. 2b). This type contains a branching node connected to multiple nodes. The rule $isCitizenOf(x, y) \leftarrow isMarriedTo(x, z_1) \wedge isCitizenOf(z_1, z_2) \wedge DealsWith(z_1, y)$ branches at node z_1, which connects to both z_2 and y. This rule captures citizenship through familial and relational dependencies.

Edge-Cyclic Rule Body (Fig. 2c). This type contains multiple edges between the same entity pair. The rule $isCitizenOf(x, y) \leftarrow LivesIn(x, y) \wedge WorksAt(x, y)$ forms an edge cycle in which two relations connect x and y, showing how multiple co-location signals support citizenship inference.

Node-Cyclic Rule Body (Fig. 2d). This type contains shared intermediate nodes that form cycles. The rule $isLocatedIn(x, y) \leftarrow hasCapital(x, z_1) \wedge hasCapital(y, z_1) \wedge hasOfficialLanguage(x, z_2) \wedge hasOfficialLanguage(y, z_2)$ connects x and y through the shared capital z_1 and language z_2, reflecting how shared attributes can indicate hierarchical relationships.

Appendix E Multi-seed Robustness Analysis

We evaluate GRiD under five random seeds to assess robustness. As shown in Table 6, the results remain stable across datasets. The 95% confidence intervals are small, and the variance stays close to zero

Table 5: Representative chain-like and graph-like rules learned by GRiD on YAGO3-10.

Rule Type	Rule Head	Rule Body
Chain-like	isLocatedIn	$\text{hasNeighbor}(x, z) \wedge \text{isLocatedIn}(z, y)$
		$\text{hasNeighbor}(x, z_1) \wedge \text{DealsWith}(z_1, z_2) \wedge \text{isLocatedIn}(z_2, y)$
		$\text{hasCapital}(x, z) \wedge \text{participatedIn}(x, z) \wedge \text{happenedIn}(z, y)$
	isCitizenOf	$\text{LivesIn}(x, y)$
		$\text{isMarriedTo}(x, z) \wedge \text{isCitizenOf}(z, y)$
		$\text{hasAcademicAdvisor}(x, z_1) \wedge \text{worksAt}(z_1, z_2) \wedge \text{isLocatedIn}(z_2, y)$
Graph-like	isLocatedIn	$\text{hasCapital}(x, z_1) \wedge \text{hasCapital}(y, z_1) \wedge \text{hasOfficialLanguage}(x, z_2) \wedge \text{hasOfficialLanguage}(y, z_2)$
		$\text{exports}(x, z_1) \wedge \text{imports}(y, z_1) \wedge \text{hasCurrency}(x, z_2) \wedge \text{hasCurrency}(y, z_2)$
		$\text{hasNeighbor}(x, z_1) \wedge \text{hasNeighbor}(y, z_1) \wedge \text{exports}(x, z_2) \wedge \text{imports}(y, z_2)$
	isCitizenOf	$\text{LivesIn}(x, y) \wedge \text{WorksAt}(x, y)$
		$\text{isMarriedTo}(x, z_1) \wedge \text{isCitizenOf}(z_1, z_2) \wedge \text{DealsWith}(z_1, y)$
		$\text{GraduatedFrom}(x, z_1) \wedge \text{isLocatedIn}(z_1, y) \wedge \text{hasAcademicAdvisor}(x, z_2) \wedge \text{livesIn}(z_2, y)$

Table 6: Performance evaluation results under different random seeds across various datasets.

Seed	Family			Kinship			UMLS		
	MRR (%)	HIT1 (%)	HIT10 (%)	MRR (%)	HIT1 (%)	HIT10 (%)	MRR (%)	HIT1 (%)	HIT10 (%)
42	95.38	91.75	99.64	68.33	56.46	92.43	83.45	74.96	95.60
123	94.97	91.11	99.61	67.02	53.90	92.67	84.01	75.72	97.12
456	95.31	91.60	99.71	66.26	52.27	92.67	83.57	74.66	96.81
789	95.20	91.38	93.00	67.14	54.48	92.43	82.93	73.97	97.11
1024	95.39	91.86	99.64	67.17	54.71	92.08	83.43	75.11	95.75
MEAN	95.25	91.54	98.32	67.18	54.36	92.46	83.48	74.89	96.48
95% CI	±0.22	±0.37	±3.69	±0.92	±1.88	±0.30	±0.48	±0.80	±0.93
VAR	0.00	0.00	0.09	0.00	0.02	0.00	0.35	0.57	0.67

Seed	WN-18RR			FB15K-237			YAGO3-10		
	MRR (%)	HIT1 (%)	HIT10 (%)	MRR (%)	HIT1 (%)	HIT10 (%)	MRR (%)	HIT1 (%)	HIT10 (%)
42	45.27	40.26	54.66	42.31	32.32	61.72	55.70	50.42	63.80
123	43.91	39.40	52.50	42.61	32.16	63.04	56.78	51.64	65.23
456	44.24	39.80	52.73	42.41	32.04	62.51	52.53	47.31	60.66
789	42.73	38.30	51.09	42.63	32.33	62.18	54.93	49.86	62.71
1024	42.74	38.33	50.85	41.49	31.40	61.17	54.62	49.65	62.47
MEAN	43.78	39.22	52.37	42.29	32.05	62.13	54.91	49.78	62.97
95% CI	±1.34	±1.09	±1.90	±0.58	±0.48	±0.89	±1.95	±1.96	±2.10
VAR	0.01	0.01	0.02	0.42	0.34	0.64	1.41	1.41	1.51

Table 7: Statistics and properties of the experimental datasets.

Dataset	# Rel	BF_Schema	Gini	NPIM	N-to-N ratio	Chain Better	Graph Better
Family	12	7.02	0.20	0.55	66.70%	100.00%	0.00%
Kinship	25	30.22	0.39	0.36	88.00%	72.00%	0.00%
UMLS	46	11.48	0.64	0.36	67.40%	30.43%	17.39%
WN-18RR	11	2.34	0.67	-0.07	18.20%	63.64%	27.27%
FB15K-237	237	10.33	0.68	0.47	45.60%	68.78%	20.68%
YAGO3-10	37	2.88	0.83	0.26	45.90%	70.27%	27.03%

on small datasets and below 1.5 on more complex datasets. These results indicate that GRiD is not sensitive to random initialization.

Appendix F KGC Pipeline

We apply generated rules to KGC through sparse matrix grounding and filtered ranking, illustrated in Algorithm 1. For each relation $r \in \mathcal{R}$, let $\mathbf{M}_r \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{E}|}$ denote its sparse adjacency matrix, and let $\mathbf{M}_r^{\text{gt}}$ denote the ground-truth matrix used for scoring and filtering. The generated rules for r are partitioned into Ω_r^{chain} and Ω_r^{graph}. For a rule ρ, $\text{Ground}(\rho)$ returns its body grounding matrix: chain-like rules are grounded by sparse matrix multiplication,

Algorithm 1 KGC with Generated Rules

Require: Matrices $\{\mathbf{M}_r, \mathbf{M}_r^{\text{gt}}\}_{r\in\mathcal{R}}$, rules $\{\Omega_r^{\text{chain}}, \Omega_r^{\text{graph}}\}_{r\in\mathcal{R}}$, fusion weight α, queries Q
Ensure: Filtered ranks $\{\text{rank}_q\}_{q\in Q}$

```
for r ∈ R do
    S_r^chain, S_r^graph ← 0, 0
    for ρ ∈ Ω_r^chain ∪ Ω_r^graph do
        M_ρ ← Ground(ρ)
        s(ρ) ← Score(M_ρ, M_r^gt)
        S_r^type(ρ) ← S_r^type(ρ) + s(ρ) M_ρ
    end for
    S_r ← (1 − α) S_r^chain + α S_r^graph
end for
for q = (e_h, r, e_t^⋆) ∈ Q do
    s_q ← S_r[e_h, :]
    Mask all known valid tails except e_t^⋆ in s_q
    rank_q ← 1 + Σ_{e∈E} 𝕀(s_q[e] > s_q[e_t^⋆])
end for
return {rank_q}_{q∈Q}
```

while graph-like rules are grounded by decomposing the body into chain components and taking their element-wise conjunction. $\text{Score}(\cdot)$ computes rule quality from coverage, confidence, and PCA-confidence. For each query $(e_h, r, ?)$, candidate tail entities are ranked by $\mathbf{S}_r[e_h, :]$ under the filtered setting. We report Hit@K and MRR: $\text{Hit@K} = \frac{1}{|Q|}\sum_{q\in Q}\mathbb{I}(\text{rank}_q \le K)$, $\text{MRR} = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{\text{rank}_q}$.

Appendix G Rule Dataset Construction

For supervised pre-training, we construct rule instances from KGs in two steps. First, for each training triple (e_h, r_h, e_t), we sample fixed-length random walks from e_h and convert the resulting relation paths into chain-like rule bodies by replacing entities with logical variables. Second, each chain is augmented once into a graph-like rule by sampling one topology from Fig. 2 and adding the required auxiliary variables and relation labels. Each constructed instance is represented by a rule-body adjacency matrix $\mathbf{A}_0$, with the target relation r_h used as the generation condition.