# DICE: A Dynamic Subgraph and Interaction–Type Collaborative Enhancement Framework for One-shot Subgraph Reasoning

1st Zhiyong Wang
*School of Computer Science and Technology, Xinjiang University*
Urumqi, China
2510602522@qq.com

2nd Di Ying
*School of Computer Science and Technology, Xinjiang University*
Urumqi, China
yd77889977@163.com

3rd Hui Zhao*
*School of Computer Science and Technology, Xinjiang University*
Urumqi, China
zhaohui@xju.edu.cn

***Abstract*—Large-scale knowledge graph completion (KGC) must balance inference efficiency and prediction reliability. Although one-shot subgraph reasoning greatly reduces computation, existing methods often use fixed-budget heuristic sampling that cannot adapt to query-specific structural distributions, causing either insufficient evidence coverage or redundant noise. Moreover, ranking candidates mainly by subgraph propagation may yield false positives that are structurally reachable but semantically or type-inconsistent, degrading top-rank quality. We propose a dynamic subgraph and interaction–type collaborative enhancement framework for one-shot subgraph reasoning(DICE). DICE determines subgraph size via a cumulative Personalized PageRank (PPR) mass threshold with a minimum-size constraint and key-node retention. On top of a replaceable subgraph reasoning backbone, DICE adds an *Entity Interaction Module* (EIM) for relation-conditioned interaction patterns and a *Type-Aware Constraint* (TAC) for tail-type preferences, with a gating mechanism to fuse evidence adaptively. Experiments on WN18RR, NELL-995, and YAGO3-10 show that DICE outperforms strong baselines; ablations verify complementary gains; and subgraph/backbone tests demonstrate robustness and transferability.**

***Index Terms*—Knowledge graph completion, one-shot subgraph reasoning, personalized PageRank, entity interaction, type-aware constraint**

## I. Introduction

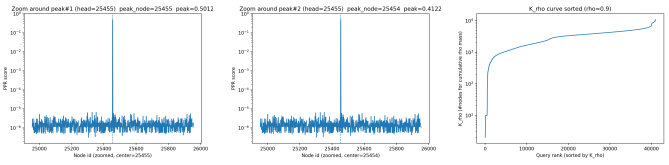
Knowledge graph completion (KGC) predicts missing facts in incomplete knowledge graphs and supports downstream applications such as question answering, retrieval, recommendation, and scientific discovery. In practice, KGs are constructed from heterogeneous sources and inevitably suffer from incompleteness and noise, making reliable and efficient link prediction a long-standing core problem.

Existing KGC approaches can be roughly grouped into: (1) *Embedding-based scoring*, which learns entity/relation representations and scores triples with parametric functions (e.g., TransE [1], DistMult [2], ComplEx [3], RotatE [4], ConvE [5]). They are efficient at inference but often rely on large entity embedding tables and may struggle with long-tail entities, compositional relations, or scenarios requiring explicit multi-hop evidence; (2) *Rule/path reasoning*, which searches or learns relational paths/rules (e.g., PRA [6], NeuralLP [7], MINERVA [8]), but may face path explosion or unstable search/training on large graphs; (3) *GNN-style relational reasoning*, which aggregates local evidence with relation-aware message passing (e.g., R-GCN [9]), yet full-graph propagation can be costly on large KGs; (4) *Subgraph-based reasoning*, which restricts computation to query-related neighborhoods to balance evidence richness and efficiency (e.g., GraIL [10], NBFNet [11], RED-GNN [15]).

Subgraph-based reasoning alleviates scalability by limiting computation to query-relevant areas. However, many methods still require repeated subgraph extraction per candidate (e.g., enclosing subgraphs in GraIL [10]), which is expensive at inference. Less is More (LiM) introduced *one-shot subgraph reasoning* [12]: it extracts a query-dependent subgraph once per query and ranks candidates within it, where Personalized PageRank(PPR) [13] provides an efficient localization heuristic. Despite its efficiency, two challenges remain: (i) **fixed-budget sampling is brittle across queries**: Top-$K$ sampling cannot adapt to query-specific PPR shapes. Sharp distributions introduce redundant noise, while flat ones risk insufficient evidence coverage (Fig. 1); (ii) **structural reachability does not guarantee semantic feasibility**: ranking candidates mainly by subgraph propagation may yield false positives that are structurally reachable but relation/type-inconsistent, degrading top-rank quality.

We propose a dynamic subgraph and interaction–type collaborative enhancement framework for one-shot subgraph reasoning(**DICE**). First, DICE determines the subgraph size by a *cumulative PPR mass coverage* threshold with a minimum-size constraint. In addition, we introduce a key-node retention mechanism: beyond probability-mass coverage, we enforce a minimum subgraph size to ensure that the extracted subgraph retains a sufficient number of nodes for effective reasoning. This stabilizes evidence coverage across queries with diverse PPR distribution patterns and reduces the risk of information

*Corresponding authors.
This article is supported by the Key Research and Development Program of Xinjiang Uygur Autonomous Region (Project No. 2023B01032).



(a) example 1 (b) example 2 (c) Global ($\rho$=0.9)

Fig. 1. PPR distributions exhibit disparities both locally and globally. (a) and (b) show significant variations in local PPR scores across different queries; (c) reports the minimal nodes required to reach 90% cumulative PPR mass on WN18RR.

loss caused by overly small subgraphs. Second, on top of a replaceable subgraph reasoning backbone, DICE adds two lightweight and complementary signals: an *Entity Interaction Module* (EIM) that learns relation-conditioned interaction patterns to enhance structural evidence, and a *Type-Aware Constraint* (TAC) that models relation-specific tail-type preferences to calibrate semantic feasibility. A gating mechanism fuses structural and type evidence adaptively, improving robustness under noisy neighborhoods and long-tail entities.

Our contributions are:

- **Coverage-driven dynamic subgraph construction.** We adaptively determine subgraph size via cumulative PPR mass coverage, with minimum-size and key-node retention for stability.
- **Two complementary enhancement modules.** EIM adds relation-conditioned interaction evidence; TAC provides type-feasibility calibration to suppress inconsistent candidates.
- **An end-to-end one-shot framework.** We unify dynamic construction, replaceable backbones, and gated multi-branch fusion to improve robustness without sacrificing efficiency.

## II. Related Work

### A. Embedding-based KGC

Embedding-based methods learn entity/relation representations and score triples efficiently, such as TransE [1], DistMult [2], ComplEx [3], RotatE [4], ConvE [5] and QuatE [14]. They are computationally attractive at inference, but can be less reliable for long-tail entities or queries that require explicit multi-hop evidence.

### B. Rule/Path Reasoning

Rule- and path-based methods reason by mining or learning relational paths, including PRA [6], NeuralLP [7], and MINERVA [8]. While they provide interpretable evidence, they may suffer from path-space explosion or unstable search/training on large graphs. Recent neural-symbolic approaches such as RulE [21] further soften logical constraints by jointly modeling rules with entities and relations, but they still rely on predefined rule structures.

### C. GNN-based Relational Reasoning

Relational GNNs aggregate neighborhood evidence with relation-aware message passing, e.g., R-GCN [9]. Subsequent KGC models further improve relation modeling and propagation efficiency (e.g., CompGCN [16]). Recent work such as ULTRA [20] moves toward more transferable relation representations and stronger generalization across graphs. However, full-graph propagation can be expensive for large-scale KGs, motivating locality-restricted reasoning.

### D. Subgraph-based Methods

Subgraph reasoning restricts computation to query-related neighborhoods, providing a middle ground between efficient embedding scoring and evidence-rich reasoning. GraIL [10] extracts enclosing subgraphs per candidate; NBFNet [11] aggregates path evidence; RED-GNN [15] performs relation-aware propagation, but repeated extraction can be costly. One-shot subgraph reasoning (LiM) extracts once per query and ranks within the subgraph [12], with PPR as an effective localization heuristic. However, fixed-budget sampling and purely structural ranking can be unstable across queries and may admit type-inconsistent false positives. Our work tackles these bottlenecks via coverage-driven dynamic PPR budgeting and interaction–type collaborative enhancement for more robust one-shot reasoning.

## III. Methodology

### A. Problem Definition

A knowledge graph (KG) is often viewed as a set of triples, denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where each triple is $(h, r, t)$ with head entity $h \in \mathcal{E}$, relation $r \in \mathcal{R}$, and tail entity $t \in \mathcal{E}$. To fully exploit structured knowledge, we formalize the triple set as a directed graph $\mathbb{G}$, where vertices are entities and directed edges are relations. Each directed edge $(h_i, r_j, t_k) \in \mathbb{G}$ represents a fact. Given a query $q = (h, r, ?)$ where $h$ is the source vertex and $r$ is the relation, KGC aims to efficiently retrieve and collect a set of candidate entities $\{t_o\}$ such that $(h, r, t_o) \notin \mathbb{G}$, due to the incompleteness of the KG.

### B. Overall Framework

Fig. 2 illustrates the overall pipeline of DICE, which consists of three stages: (1) dynamic subgraph construction; (2) information aggregation; and (3) multi-branch discrimination and fusion. We describe each stage in detail below. To make the one-shot reasoning procedure more concrete, Fig. 3 visualizes the end-to-end inference for a single query, from PPR-based subgraph extraction to candidate scoring and final prediction.

### C. Stage 1: Dynamic Subgraph Construction

In the one-shot subgraph reasoning paradigm, the key is to construct an informative yet size-controlled query-related subgraph for each head entity $h$. We estimate node importance using Personalized PageRank (PPR) scores and propose a coverage-driven dynamic budget strategy to adapt to different PPR distribution shapes.

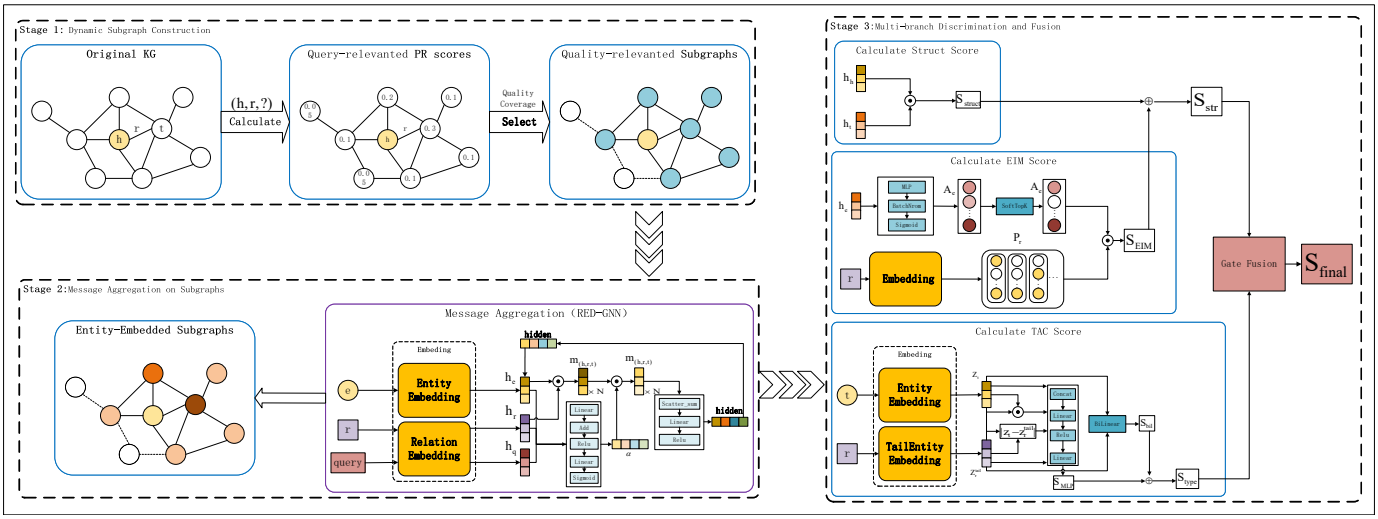



Fig. 2. The proposed framework of DICE.

*1) PPR Score Computation:* After $k$ iterations, we obtain the stationary PPR distribution vector personalized to the query head entity $h$; Here, $s \in \mathbb{R}^{|E|}$ denotes the personalization (restart) vector, i.e., a one-hot vector with $s_h = 1$ and $s_e = 0$ for all $e \neq h$:

$$p_{(k+1)} = \alpha \cdot s + (1-\alpha) \cdot D^{-1}A \cdot p_{(k)}, \tag{1}$$

$$p_{(k+1)} \triangleq p_h \in \mathbb{R}^{|\mathcal{E}|}, \quad \sum_{e \in \mathcal{E}} p_h(e) = 1. \tag{2}$$

Here $A \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{E}|}$ is the adjacency matrix, $D \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is the degree matrix, and $\alpha \in (0,1)$ is the restart probability. $p_h(e)$ is the steady-state probability of visiting entity $e$ in a restart random walk starting from $h$, and it can be regarded as the structural relevance of entity $e$ to the current query.

*2) Dynamic Budget via Cumulative Mass Coverage:* To adapt to different PPR distributions, instead of a fixed Top-$K$ budget we determine subgraph size by cumulative probability mass. We sort all entities by PPR scores in descending order and define the cumulative mass of the top-$k$ entities $M_h(k)$:

$$p_h(e_1) \geq p_h(e_2) \geq \cdots \geq p_h(e_{|\mathcal{E}|}), \quad M_h(k) = \sum_{i=1}^{k} p_h(e_i). \tag{3}$$

Given a mass-coverage ratio $\rho \in (0,1)$, we choose the smallest $k$ satisfying $M_h(k) \geq \rho$ as the dynamic budget; to ensure stability and usability, we impose a minimum node constraint $K_{\min}$ to prevent overly small subgraphs:

$$K(h) = \min\{k : M_h(k) \geq \rho\}, \tag{4}$$

$$K(h) \leftarrow \max\{K(h), K_{\min}\}. \tag{5}$$

This strategy is naturally adaptive: when $p_h$ is "sharp" (mass highly concentrated), a small $K(h)$ already covers most mass, producing a compact and evidence-focused subgraph; when $p_h$ is "flat" (mass dispersed), a larger $K(h)$ is required to reach the same coverage, thus automatically expanding the subgraph to improve evidence coverage and reduce missing-recall risk.

*3) Subgraph Construction:* After node selection, we obtain the subgraph entity set $\mathcal{E}_s(h)$. We then construct the subgraph triple set $\mathcal{T}_s(h)$ by induction, yielding the dynamic subgraph associated with the mass coverage:

$$\mathcal{E}_s(h) = \{h\} \cup \{e_1, e_2, \ldots, e_{K(h)}\}, \tag{6}$$

$$\mathcal{T}_s(h) = \{(x, r, y) \in \mathcal{T} \mid x \in \mathcal{E}_s(h),\ y \in \mathcal{E}_s(h)\}, \tag{7}$$

$$\mathcal{G}_s(h) = (\mathcal{E}_s(h), \mathcal{R}, \mathcal{T}_s(h)). \tag{8}$$

In subsequent stages, we perform structural reasoning and candidate ranking only on $\mathcal{G}_s(h)$, retaining the efficient one-shot paradigm.

### D. Stage 2: Information Aggregation

Given the dynamic subgraph $\mathcal{G}_s(h)$, the goal of this stage is to perform relation-aware aggregation on the subgraph to learn contextual representations for nodes and produce structural-branch scores. We abstract this part as a replaceable reasoner:

$$H^L = \text{GNN\_MODEL}\big(\mathcal{G}_s(h)\big), \tag{9}$$

where $H^L$ denotes the final-layer node embeddings. Our default implementation follows RED-GNN-style layer-wise expansion and aggregation (Fig. 2, Stage 2; more details are provided in Appendix B), but it can be replaced by other multi-relational GNNs (e.g., CompGCN [16]) as long as the interface (output node representations) remains consistent.

Although we adopt a RED-GNN-style reasoner by default, we design it as a *replaceable backbone*. EIM depends only on the final-layer embeddings $H^L$ and performs relation-conditioned matching in an interaction space, so it can be plugged into any encoder that outputs node representations. TAC judges consistency between entity type vectors and relation tail-type prototypes and is independent of the propagation mechanism. Thus, DICE can be viewed as "dynamic subgraph construction + a subgraph reasoning backbone + weakly-coupled enhancement heads", facilitating flexible backbone choices and easy transfer.

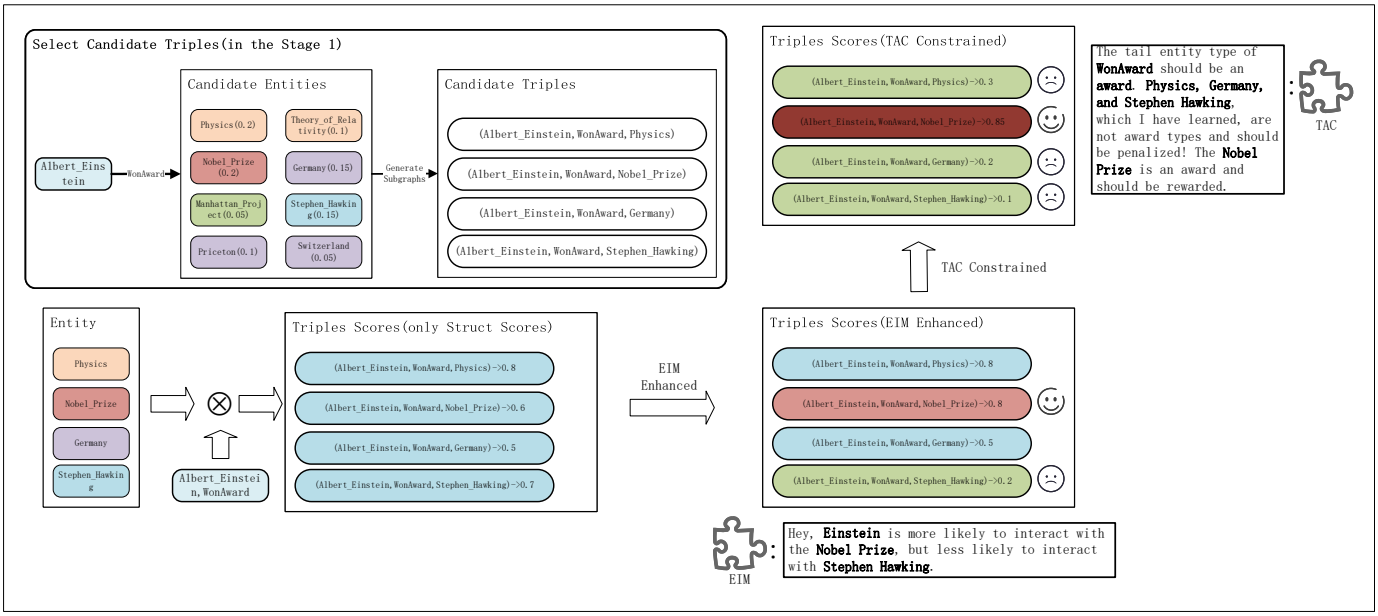



Fig. 3. Schematic diagram of the one-shot subgraph reasoning process.

### *E. Stage 3: Multi-branch Discrimination and Fusion*

In addition to the structural branch, we introduce two complementary discrimination branches: EIM and the TAC, and perform hierarchical fusion to obtain the final score $S_{\text{final}}(h, r, t)$. The structural branch provides local structural evidence; EIM provides relation-conditioned latent interaction evidence to complement structural expressiveness; and TAC provides semantic-feasibility constraints to calibrate and suppress inconsistent candidates.

*1) EIM:* EIM models latent interaction patterns between entities under a relation condition. Given the final-layer entity representation $h_e$, we first map it into a $K$-dimensional *interaction pattern space* and obtain a non-negative response vector (an interaction probability signature):

$$a_e = f_{\text{proj}}(h_e) \in [0,1]^K, \quad e \in \{h, t\}, \tag{10}$$

where $f_{\text{proj}}(\cdot)$ is an MLP followed by a sigmoid.

To highlight a small number of key patterns, we use a learnable soft-Top$k$ mechanism to generate a mask $m_e \in [0,1]^K$ and obtain a sparse signature $\widetilde{a}_e$:

$$m_e = \text{SoftTopK}(a_e, k_{\text{eff}}, T), \quad \widetilde{a}_e = a_e \odot m_e, \tag{11}$$

where $k_{\text{eff}}$ is the expected number of key patterns, $T$ is a temperature, and $\odot$ is the Hadamard product. $\text{SoftTopK}(\cdot, k_{\text{eff}}, T)$ returns a (continuous) mask that preserves the indices of the top-$k_{\text{eff}}$ entries of the input scores (with softness controlled by $T$), assigning values close to 1 to selected entries and close to 0 to the others.

After obtaining the sparse signatures $\widetilde{a}_h, \widetilde{a}_t$, we further introduce a relation-conditioned *pattern transition/interaction matrix* to capture interaction regularities under different relations. For each relation $r \in \mathcal{R}$, we learn a matrix $P_r \in \mathbb{R}^{K \times K}$:

$$P_r = \tanh\big(\text{reshape}(E_p(r))\big), \tag{12}$$

where $E_p(\cdot)$ is an embedding function producing $E_p(r) \in \mathbb{R}^{K \times K}$ in a semantic space.

For a query $q = (h, r, ?)$, we map the head signature by $P_r$ and match it with the candidate tail signature to obtain the interaction score:

$$S_{\text{EIM}}(h, r, t) = \big(\widetilde{a}_h^\top P_r\big)\widetilde{a}_t. \tag{13}$$

This branch is compatible with different GNN backbones because it only requires final-layer entity embeddings.

*2) TAC:* Relying only on the structural branch and EIM may promote candidates that are "structurally reachable and interaction-similar but semantically type-inconsistent". We therefore design TAC to explicitly model *relations' tail-type preferences*, providing lightweight semantic-feasibility calibration for ranking.

We learn a type vector $z_t \in \mathbb{R}^d$ for each tail entity $t$ and an expected tail-type prototype $z_r^{\text{tail}} \in \mathbb{R}^d$ for each relation $r$. Specifically, both $z_t$ and $z_r^{\text{tail}}$ are learnable projections directly derived from the corresponding tail-entity embedding and relation embedding, respectively. Then, we adopt a "two-path" discriminator to capture both global compatibility and fine-grained interactions. The bilinear branch captures global trends, while the MLP branch models non-linear interaction features. The final type-constraint score is a weighted fusion:

$$S_{\text{bil}}(r, t) = z_t^\top W_b \, z_r^{\text{tail}}, \tag{14}$$

$$S_{\text{mlp}}(r, t) = \text{MLP}_{\text{type}}\Big([z_t, z_r^{\text{tail}}, z_t \odot z_r^{\text{tail}}, |z_t - z_r^{\text{tail}}|]\Big), \tag{15}$$

$$S_{\text{type}}(r, t) = \gamma_b \cdot S_{\text{bil}}(r, t) + \gamma_m \cdot S_{\text{mlp}}(r, t), \tag{16}$$

where $\gamma_b$ and $\gamma_m$ are learnable weights. $W_b \in \mathbb{R}^{d \times d}$ is a learnable bilinear transformation matrix used to model the compatibility between the tail-type vector $z_t$ and the relation-specific prototype $z_r^{\text{tail}}$. Since this branch directly operates on

entity/relation type representations, it is decoupled from the propagation mechanism of GNN_MODEL.

*3) Multi-score Fusion:* For candidate triples $(h, r, t)$ generated by query $q = (h, r, ?)$, the structural score is produced by a readout function (including linear readout and dot-product readout):

$$S_{\text{struct}}(h, r, t) = \text{Readout}(h_h, h_t). \tag{17}$$

Here, $h_h$, $h_t$ denote the embedding representations of the head entity $h$ and tail entity $t$, respectively, all produced by GNN_MODEL.

After obtaining three scores, we perform hierarchical fusion. We first treat the interaction score as a learnable enhancement for the structural branch to obtain an enhanced structural score $S_{\text{str}}(h, r, t)$, and then use a lightweight gate network to generate a sample-wise fusion coefficient $gate(h, r, t) \in (0, 1)$. The final score is a convex combination:

$$S_{\text{str}}(h, r, t) = S_{\text{struct}}(h, r, t) + \alpha \cdot S_{\text{EIM}}(h, r, t), \tag{18}$$

$$gate(h, r, t) = \sigma\Big(W_g^\top \left[S_{\text{str}}(h, r, t), S_{\text{type}}(r, t)\right] + b_g\Big), \tag{19}$$

$$\begin{aligned} S_{\text{final}}(h, r, t) = {} & gate(h, r, t) \cdot S_{\text{str}}(h, r, t) \\ & + (1 - gate(h, r, t)) \cdot S_{\text{type}}(r, t). \end{aligned} \tag{20}$$

Here $\alpha$ is a scaling factor, $\sigma(\cdot)$ is the sigmoid, and $W_g, b_g$ are learnable gate parameters. The gated fusion mechanism adaptively assigns instance-wise weights to the structure-enhanced and type-aware branches according to signal reliability, enabling dynamic complementary fusion that suppresses false positives and stabilizes predictions.

*4) Loss Function:* For each training sample (query) $(h, r, ?)$, the model outputs a score vector $S(h, r) \in \mathbb{R}^{|\mathcal{E}_s(h)|}$ over the candidate entity set extracted from the sampled subgraph $G_s(h)$, where the $t$-th entry $S(h, r)[t]$ is the predicted score for a candidate tail entity $t \in \mathcal{E}_s(h)$. We use a LogSumExp (LSE) multi-class ranking loss $L_{\text{rank}}(h, r, t^+)$. To mitigate overfitting of the interaction matrices, we apply an $\ell_2$-Frobenius regularization on $P_r$ in EIM, yielding $L_{\text{EIM}}$. For a batch $\mathcal{B}$, the total loss is:

$$L = \sum_{(h,r,t^+)\in\mathcal{B}} L_{\text{rank}}(h, r, t^+) + L_{\text{EIM}}, \tag{21}$$

$$L_{\text{rank}}(h, r, t^+) = -s^+ + \log \sum_{t\in\mathcal{E}_s(h)} \exp\big(S(h, r)[t]\big), \tag{22}$$

$$L_{\text{EIM}} = \lambda \cdot \|P_r\|_F^2, \tag{23}$$

where $s^+ = S(h, r)[t^+]$ is the score of the ground-truth tail entity, and $\lambda$ is a hyper-parameter (default $0.5$).

## IV. Experiments

### A. Experimental Setup

**Datasets.** We evaluate on WN18RR, NELL-995, and YAGO3-10, covering lexical relations, noisy automatically constructed KBs, and large-scale multi-domain graphs. Detailed dataset statistics are provided in Appendix A.

**Training and inference environment.** All experiments are run on a single-machine server with an NVIDIA RTX 4090 GPU, using a unified training pipeline and the same evaluation scripts to ensure comparability.

**Evaluation Metrics.** Following standard KGC practice, we evaluate via entity ranking. For each test triple $(h, r, t)$, we perform tail prediction by ranking all candidate entities given head $h$ and relation $r$. We report Mean Reciprocal Rank (MRR) and Hits@N, and we focus on Hits@1 and Hits@10 in our experiments;The higher MRR, Hits@1 and Hits@10 means the better result.

**Implementation Details.** We implement our model using the Adam optimizer. The hidden dimension is set to 256 for WN18RR, 128 for NELL-995, and 64 for YAGO3-10. We use 8 GNN layers across all datasets. The learning rate is chosen as 0.0001 on WN18RR, 0.0011 on NELL-995, and 0.0010 on YAGO3-10. For the entity interaction module (EIM), we set the pattern dimension $K$ to 128, 256, and 256 respectively, with effective pattern number $K_{\text{eff}}$ set to 10, 50, and 20. The regularization weight $\lambda$ for EIM is fixed to 0.5. In PPR computation, the restart probability $\alpha$ is set to 0.1 and the cumulative mass threshold $\rho$ is set to 0.9 for all datasets.

### B. Main Results and Analysis

We compare DICE with representative baselines from embedding methods, path/rule methods, and structural/subgraph reasoning methods on WN18RR, NELL-995, and YAGO3-10. Results are summarized in Table I. Overall, DICE achieves the best performance on all three datasets. Compared with the representative one-shot subgraph method One-Shot-Subgraph [12], DICE provides consistent gains: on WN18RR it improves MRR/H@1/H@10 by 0.9/1.2/0.9 points; on NELL-995 by 0.9/1.6/1.1; and on YAGO3-10 by 0.7/0.6/0.9. These results indicate that, under the efficient one-shot paradigm (extract once + rank within subgraph), injecting relation-conditioned interaction evidence and type-feasibility constraints effectively alleviates insufficient evidence and unstable ranking in open-world settings, yielding stable and transferable improvements. Rule-reasoning baselines include DRUM [17] and RNNLogic [18]; structural reasoning baselines include DPMPN [19] and the one-shot baseline One-Shot-Subgraph [12].

### C. Ablation Experiments

To validate the effectiveness and necessity of each component, we conduct ablation studies on WN18RR and YAGO3-10 (Table II). Starting from the full model (DICE), we remove EIM (w/o EIM), remove TAC (w/o TAC), and remove both (w/o EIM&TAC), while keeping other settings, subgraph construction strategy, and evaluation procedures unchanged. Removing either module consistently degrades performance, and removing both causes the largest drop, suggesting the improvements mainly come from the two enhancement branches and their fusion rather than random training noise or hyper-parameter bias. The full model remains best on both datasets, indicating stability across distributions.

TABLE I
EMPIRICAL RESULTS ON WN18RR, NELL-995, AND YAGO3-10. BEST RESULTS ARE IN **BOLD**; SECOND BEST ARE UNDERLINED. "–" INDICATES UNAVAILABLE RESULTS.

| Model | WN18RR | | | NELL-995 | | | YAGO3-10 | | |
|---|---|---|---|---|---|---|---|---|---|
| | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 |
| ConvE | 42.7 | 39.2 | 49.8 | 51.1 | 44.6 | 61.9 | 52.0 | 45.0 | 66.0 |
| QuatE | 48.0 | 44.0 | 55.1 | 53.3 | 46.6 | 64.3 | 37.9 | 30.1 | 53.4 |
| RotatE | 47.7 | 42.8 | 57.1 | 50.8 | 44.8 | 60.8 | 49.5 | 40.2 | 67.0 |
| MINERVA | 44.8 | 41.3 | 51.3 | 51.3 | 41.3 | 63.7 | – | – | – |
| DRUM | 48.6 | 42.5 | 58.6 | 53.2 | 46.0 | **66.2** | 53.1 | 45.3 | 67.6 |
| RNNLogic | 48.3 | 48.3 | 55.8 | 41.6 | 36.3 | 47.8 | 55.4 | 50.9 | 62.2 |
| CompGCN | 47.9 | 44.3 | 54.6 | 46.3 | 38.3 | 47.8 | 48.9 | 39.5 | 58.2 |
| DPMPN | 48.2 | 44.4 | 55.8 | 51.3 | 45.2 | 61.5 | 55.3 | 48.4 | 67.9 |
| NBFNet | 55.1 | 49.7 | 66.6 | 52.5 | 45.1 | 63.9 | 55.0 | 47.9 | 68.3 |
| RED-GNN | 53.3 | 48.5 | 62.4 | 54.3 | 47.6 | 65.1 | 55.9 | 48.3 | 68.9 |
| RulE | 51.9 | 47.5 | 60.5 | – | – | – | 53.5 | 44.7 | 69.4 |
| ULTRA | 55.1 | – | 66.6 | 54.3 | – | 65.1 | 56.3 | – | 70.8 |
| One-Shot-Subgraph | 56.7 | 51.4 | 66.6 | 54.7 | 48.5 | 65.1 | 60.6 | 54.0 | 72.1 |
| DICE | **57.6** | **52.6** | **67.5** | **55.6** | **49.2** | **66.2** | **61.3** | **54.6** | **73.0** |

TABLE II
ABLATION RESULTS ON WN18RR AND YAGO3-10. MRR, HITS@1 AND HITS@10 ARE HIGHER THE BETTER, WHILE INFERENCE TIME ARE LOWER THE BETTER.

| Model | WN18RR | | | | YAGO3-10 | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | H@1 | H@10 | Time | MRR | H@1 | H@10 | Time |
| DICE w/o EIM | 57.2 | 52.2 | 66.8 | 273s | 61.0 | 54.4 | 72.4 | 3586s |
| DICE w/o TAC | 57.3 | 51.7 | 67.0 | 275s | 60.9 | 54.2 | 72.8 | 3490s |
| DICE w/o EIM&TAC | 56.7 | 51.4 | 66.6 | 266s | 60.6 | 54.0 | 72.1 | 3396s |
| DICE (full) | **57.6** | **52.6** | **67.5** | 287s | **61.3** | **54.6** | **73.0** | 3640s |

In terms of inference efficiency, removing EIM and/or TAC reduces computational overhead and accordingly decreases inference time, since both modules introduce lightweight computation for representation enhancement and constraint modeling. Specifically, the full model incurs a mild time overhead compared to the ablated variants, but achieves clear and consistent gains in all ranking metrics (MRR, Hits@1, Hits@10). This demonstrates that the performance improvement brought by EIM and TAC is cost-effective within acceptable computational overhead, rather than at the expense of excessive efficiency sacrifice.

Mechanistically, the modules address different error sources and are complementary. EIM learns relation-conditioned latent interaction patterns on top of backbone embeddings, providing higher-order interaction consistency that structural propagation may miss, thereby reducing ranking instability under subgraph noise or insufficient structural evidence. TAC explicitly models relations' tail-type preferences and adds a lightweight semantic feasibility constraint to suppress false positives that are "structurally reachable but type-inconsistent," improving the reliability of top-ranked candidates. By strengthening "structural relevance evidence" and "semantic feasibility evidence" respectively and using gated fusion for sample-wise trade-offs, they jointly yield more robust gains.

TABLE III
PREDICTION PERFORMANCE WITH DIFFERENT SAMPLING METHODS(PERCENTAGE)

| Sampling | WN18RR | | | YAGO3-10 | | |
|---|---|---|---|---|---|---|
| | MRR | H@1 | H@10 | MRR | H@1 | H@10 |
| Random Sampling | 3.8 | 3.7 | 4.2 | 5.5 | 4.9 | 6.3 |
| PageRank | 14.7 | 13.0 | 18.2 | 32.4 | 29.8 | 36.7 |
| Random Walk | 44.2 | 42.2 | 47.9 | 47.3 | 44.3 | 50.1 |
| Breadth-first Search | 55.9 | 51.2 | 65.2 | 57.1 | 51.7 | 66.3 |
| Personalized PageRank | 57.0 | 51.9 | 67.1 | 61.2 | 54.3 | 72.5 |
| Dynamic Subgraph (ours) | **57.6** | **52.6** | **67.5** | **61.3** | **54.6** | **73.0** |

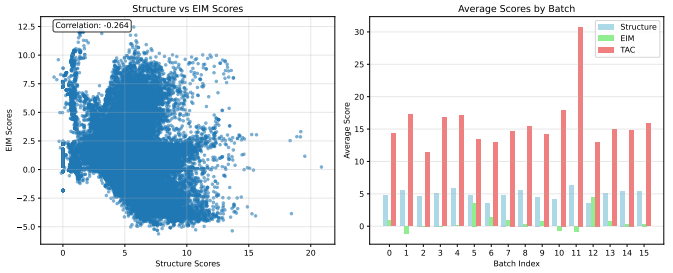



Fig. 4. Visualization of score distributions from Structure, EIM, and TAC branches on a representative inference batch. Left: scatter plot showing weak correlation between Structure and EIM signals, indicating that EIM learns non-redundant patterns. Right: bar plot showing stable per-sample contributions across queries.

### D. Analysis of Subgraph Construction Methods

To assess how subgraph construction affects the upper bound of one-shot reasoning, we fix the backbone and training setup and only vary the subgraph sampling method on WN18RR and YAGO3-10 (Table III). Results show subgraph quality is decisive: random sampling nearly fails, and PageRank performs poorly due to non-personalized centrality bias. Random Walk improves by focusing on the query neighborhood but suffers from randomness and path bias, causing noisy, unstable coverage. Breadth-first Search(BFS) further boosts performance by systematically expanding within 3–4 hops, where most evidence resides, while PPR improves again by concentrating query-centered probability mass on salient relation paths. Our dynamic subgraph performs best on both datasets, demonstrating a more robust balance between evidence coverage and redundant noise.

### E. Visualization of Module Effectiveness

To further verify the interpretability and complementary roles of EIM and TAC, we visualize the score distributions of the *Structure branch*, *EIM branch*, and *TAC branch* on a representative inference batch (Figure 4). The visualization includes a correlation scatter plot and per-sample average scores, which jointly demonstrate the effectiveness of each module.

As shown in Figure 4: (1) TAC scores present the widest distribution and strongest discriminative power to distinguish consistent/inconsistent entities, where structure scores offer

TABLE IV
PERFORMANCE WITH DIFFERENT BACKBONES AND PLUG-IN SETTINGS (PERCENTAGE).

| Model | WN18RR | | | YAGO3-10 | | |
|---|---|---|---|---|---|---|
| | MRR | H@1 | H@10 | MRR | H@1 | H@10 |
| CompGCN | 47.9 | 44.3 | 54.6 | 48.9 | 39.5 | 58.2 |
| CompGCN + One-Shot-Subgraph | 50.1 | 45.9 | 57.8 | 51.5 | 47.3 | 60.5 |
| CompGCN + DICE | 50.7 | 46.6 | 58.3 | 52.1 | 47.9 | 61.3 |
| RED-GNN | 53.3 | 48.5 | 62.4 | 55.9 | 48.3 | 68.9 |
| RED-GNN + One-Shot-Subgraph | 56.7 | 51.4 | 66.6 | 60.6 | 54.0 | 72.1 |
| RED-GNN + DICE | **57.6** | **52.6** | **67.5** | **61.3** | **54.6** | **73.0** |

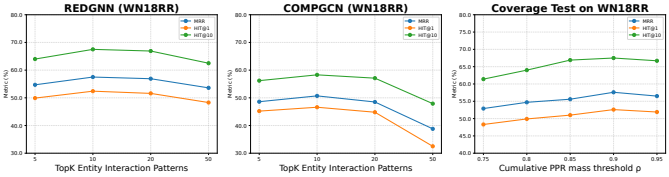



Fig. 5. Sensitivity analysis of the framework with respect to the number of interaction patterns $K_{\text{eff}}$ and the cumulative PPR mass threshold $\rho$ on knowledge graph completion. Results are shown on WN18RR. Left: RED-GNN backbone. Middle: CompGCN backbone. Right: subgraph coverage threshold $\rho$.

stable basic evidence and EIM scores act as fine-grained modifiers. (2) The weak negative correlation ($\boldsymbol{\rho = -0.264}$) between Structure and EIM scores proves EIM captures non-redundant interaction patterns independent of structure. (3) Per-batch averages verify the three branches make stable and complementary contributions for reasoning robustness. These observations validate that EIM and TAC effectively compensate for pure structural reasoning, consistent with ablation study gains.

### *F. Backbone Compatibility and Plug-in Effectiveness*

We evaluate backbone compatibility with two representative GNN backbones (CompGCN and RED-GNN) on WN18RR and YAGO3-10 (Table IV). We compare three settings: Backbone, Backbone + One-Shot-Subgraph, and Backbone + DICE, where DICE adds dynamic subgraph construction and multi-branch enhancement/fusion while keeping other settings unchanged. We report MRR, Hits@1, and Hits@10.

Results show that the one-shot paradigm consistently improves both backbones, and DICE further yields additional gains, indicating transferable improvements rather than backbone-specific tuning. The dynamic subgraph adjusts its size via a coverage threshold (smaller for sharp PPR, larger for flat PPR), improving evidence coverage and signal-to-noise ratio, while multi-branch scoring and gated fusion further suppress type-inconsistent or noisy candidates. Table III highlights the impact of higher-quality dynamic subgraphs, and Table IV confirms the plug-in effectiveness across backbones.

### *G. Sensitivity to Interaction Pattern Numbers and Subgraph Coverage*

As shown in Fig. 5, we analyze the sensitivity of EIM with respect to both the number of effective interaction patterns $K_{\text{eff}}$ and the cumulative PPR mass threshold $\rho$ used for subgraph selection. Left figure and middle figure show that increasing $K_{\text{eff}}$ from 5 to 10 improves MRR and Hits@1/10 for both backbones, while larger values (20 or 50) lead to saturated or degraded performance, indicating an optimal moderate $K_{\text{eff}}$. This aligns with WN18RR: a small $K_{\text{eff}}$ underfits relation-conditioned interactions, whereas an overly large $K_{\text{eff}}$ introduces noisy patterns, making signatures overly dense and potentially amplifying subgraph noise. Right figure examines the effect of the cumulative PPR mass threshold $\rho$ on subgraph construction. A small $\rho$ yields insufficient subgraphs that fail to support effective interaction modeling, whereas an overly large $\rho$ introduces substantial subgraph noise, which in turn degrades the effectiveness of subgraph inference.

## V. CONCLUSION

We propose DICE, a dynamic subgraph and interaction–type collaborative enhancement framework for one-shot subgraph reasoning in KGC. DICE stabilizes evidence localization via coverage-driven dynamic PPR budgeting with minimum size and key-node retention, and enhances ranking with relation-conditioned interaction modeling (EIM) and type-feasibility calibration (TAC) fused by a gate. Experiments on WN18RR, NELL-995, and YAGO3-10 show state-of-the-art performance, while ablations and cross-backbone tests verify complementarity, robustness, and transferability.

## REFERENCES


[1] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data. In: *Advances in Neural Information Processing Systems* (NeurIPS) (2013)

[2] Yang, B., Yih, W.-T., He, X., Gao, J., Deng, L.: Embedding Entities and Relations for Learning and Inference in Knowledge Bases (DistMult). In: *International Conference on Learning Representations* (ICLR) (2015). arXiv:1412.6575

[3] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex Embeddings for Simple Link Prediction (ComplEx). In: *International Conference on Machine Learning* (ICML) (2016). arXiv:1606.06357

[4] Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In: *International Conference on Learning Representations* (ICLR) (2019). arXiv:1902.10197

[5] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D Knowledge Graph Embeddings (ConvE). In: *AAAI Conference on Artificial Intelligence* (AAAI) (2018). arXiv:1707.01476

[6] Lao, N., Mitchell, T., Cohen, W.W.: Relational Retrieval Using a Combination of Path-Constrained Random Walks. *Machine Learning* **81**(1), 53–67 (2010)

[7] Yang, F., Yang, Z., Cohen, W.W.: Differentiable Learning of Logical Rules for Knowledge Base Reasoning (NeuralLP). In: *Advances in Neural Information Processing Systems* (NeurIPS) (2017). arXiv:1702.08367

[8] Das, R., Dhuliawala, S., Zaheer, M., et al.: Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases Using Reinforcement Learning (MINERVA). In: *International Conference on Learning Representations* (ICLR) (2018). arXiv:1711.05851

[9] Schlichtkrull, M., Kipf, T.N., Bloem, P., et al.: Modeling Relational Data with Graph Convolutional Networks (R-GCN). In: *European Semantic Web Conference* (ESWC) (2018). arXiv:1703.06103

[10] Teru, K.K., Denis, E., Hamilton, W.L.: Inductive Relation Prediction by Subgraph Reasoning (GraIL). In: *International Conference on Machine Learning* (ICML) (2020). arXiv:1911.06962

[11] Zhu, Z., Zhang, Z., Xhonneux, L.-P., Tang, J.: Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction (NBFNet). In: *Advances in Neural Information Processing Systems* (NeurIPS) (2021). arXiv:2106.06935
[12] Zhou, Z., Zhang, Y., Yao, J., Yao, Q., Han, B.: Less is More: One-shot Subgraph Reasoning on Large-scale Knowledge Graphs. In: *International Conference on Learning Representations* (ICLR) (2024). arXiv:2403.10231
[13] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab (1999)
[14] Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion Knowledge Graph Embeddings (QuatE). In: *Advances in Neural Information Processing Systems* (NeurIPS) (2019). arXiv:1904.10281
[15] Zhang, Y., Yao, Q.: Knowledge Graph Reasoning with Relational Digraph (RED-GNN). In: *Proceedings of the ACM Web Conference* (WWW) (2022). arXiv:2108.06040
[16] Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.: Composition-based Multi-Relational Graph Convolutional Networks (CompGCN). In: *International Conference on Learning Representations* (ICLR) (2020). arXiv:1911.03082
[17] Sadeghian, A., Armandpour, M., Ding, P., Wang, D.Z.: DRUM: End-To-End Differentiable Rule Mining on Knowledge Graphs. In: *Advances in Neural Information Processing Systems* (NeurIPS) (2019). arXiv:1911.00055
[18] Qu, M., Chen, J.K., Xhonneux, L.-P.A.C., Bengio, Y., Tang, J.: RNN-Logic: Learning Logic Rules for Reasoning on Knowledge Graphs. In: *International Conference on Learning Representations* (ICLR) (2021). arXiv:2010.04029
[19] Xu, X., Feng, W., Jiang, Y., Xie, X., Sun, Z., Deng, Z.-H.: Dynamically Pruned Message Passing Networks for Large-scale Knowledge Graph Reasoning (DPMPN). In: *International Conference on Learning Representations* (ICLR) (2020). arXiv:1909.11334
[20] Galkin, M., Yuan, X., Mostafa, H., Tang, J., Zhu, Z.: Towards Foundation Models for Knowledge Graph Reasoning (ULTRA). In: *International Conference on Learning Representations* (ICLR) (2024). arXiv:2310.04562
[21] Tang, X., Zhu, S.-C., Liang, Y., Zhang, M.: RulE: Knowledge Graph Reasoning with Rule Embedding (RulE). In: *Findings of the Association for Computational Linguistics* (ACL) (2024). arXiv:2406.04175

## APPENDIX A
## DATASET STATISTICS

Table V summarizes WN18RR, NELL-995, and YAGO3-10. WN18RR focuses on lexical hierarchies; NELL-995 contains diverse noisy relations from web text; and YAGO3-10 combines Wikipedia facts with rich ontology.

## APPENDIX B
## RED-GNN INFORMATION AGGREGATION IN STAGE 2

In Stage 2, we adopt RED-GNN-style query-conditioned message passing on the extracted subgraph. Given $q = (h, r_q, ?)$, let $\mathbf{h}_q = \mathbf{h}_{r_q}$. At layer $\ell$, for each edge $(e, r, o)$, we compute a query-conditioned message, an edge attention weight, and update node $o$ by attention-weighted aggregation.

$$\mathbf{m}_{e\to o}^{(\ell)} = \mathbf{h}_e^{(\ell-1)} \odot \mathbf{h}_r, \tag{24}$$

$$\alpha_{ero}^{(\ell)} = \sigma\Big(\mathbf{w}_\alpha^\top \operatorname{ReLU}\big(\mathbf{W}_e\mathbf{h}_e^{(\ell-1)} + \mathbf{W}_r\mathbf{h}_r + \mathbf{W}_q\mathbf{h}_q\big)\Big). \tag{25}$$

$$\mathbf{h}_o^{(\ell)} = \phi\Big(\mathbf{W}_h \sum_{(e,r,o)\in\mathcal{E}_s} \alpha_{ero}^{(\ell)}\, \mathbf{m}_{e\to o}^{(\ell)}\Big). \tag{26}$$

*a) Remark:* $\odot$ denotes Hadamard product; $\mathbf{h}_r$ and $\mathbf{h}_q$ are embeddings of relation $r$ and query relation $r_q$. Conditioning $\alpha_{ero}^{(\ell)}$ on $\mathbf{h}_q$ enables query-aware evidence aggregation within the one-shot subgraph.

TABLE V
STATISTICS OF THE EVALUATION DATASETS.

| Dataset | Entities | Relations | Train | Valid | Test | Total |
|---|---|---|---|---|---|---|
| WN18RR | 40.9k | 11 | 59.0k | 3.0k | 3.1k | 65.1k |
| NELL-995 | 74.5k | 200 | 108.9k | 0.5k | 2.8k | 112.2k |
| YAGO3-10 | 123.1k | 37 | 1079.0k | 5.0k | 5.0k | 1089.0k |

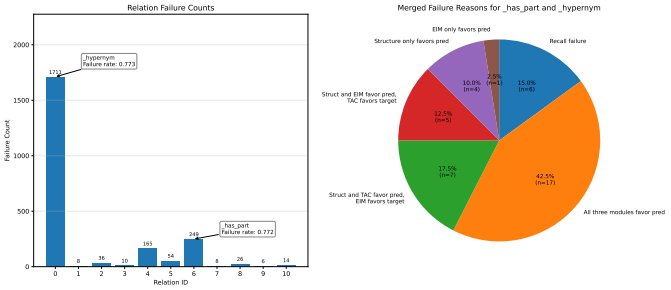



Fig. 6. Updated failure analysis on WN18RR. Left: relation-level failure counts, with "_hypernym" and "_has_part" as the two hardest relations. Right: merged error-reason distribution over failed cases from these two relations.

## APPENDIX C
## ADDITIONAL FAILURE ANALYSIS

Figure 6 shows the failure analysis on WN18RR. The left panel presents relation-level failure counts, where the major errors are concentrated on structurally complex relations, especially "_hypernym" and "_has_part", indicating that relation mapping properties strongly affect prediction difficulty. The right panel summarizes the merged error-reason distribution for these two hardest relations. Most errors fall into the category where all three branches favor the predicted tail, suggesting systematic confusion among hard candidates. Meanwhile, a non-negligible portion belongs to cases where either EIM or TAC favors the target tail while the other branches prefer the wrong one, showing that these modules provide complementary corrective signals rather than redundant information. In addition, several recall-failure cases remain, indicating that the sampled subgraph may still miss the gold entity for difficult queries. Overall, the new figure confirms both the effectiveness and the limitations of DICE: EIM and TAC can partially correct structural errors, but stronger recall-oriented sampling and finer-grained semantic discrimination are still needed for hard hierarchical relations.