PRISM: A UNIFIED HETEROGENEOUS-EXPERT AGENT FRAMEWORK FOR KNOWLEDGE GRAPH COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge graph completion (KGC) aims to infer missing facts and alleviate the inherent incompleteness of knowledge graphs. Existing KGC methods capture complementary information at different levels, including structural patterns, path and rule information, and local or global graph semantics, while multimodal KGC further incorporates textual and visual modalities. However, heterogeneous models differ in prediction reliability, output scales, and inference costs, and simply combining them cannot adaptively determine which models should be invoked or when further computation is necessary. We formulate KGC as an information-driven sequential decision problem and propose **PRISM**, a unified framework for coordinating heterogeneous reasoning models. PRISM standardizes model outputs and maintains a continuously updated shared state. The PROFILER summarizes relation, entity, and modality characteristics; the ADAPTER normalizes model outputs; the VERIFIER evaluates structural support and prediction agreement; the JUDGE performs reliability-aware rank fusion; and the ROUTER estimates the utility of additional model invocation and decides whether to CALL or STOP. Together, these components form a traceable CALL–VERIFY–STOP loop for sequential model selection and stopping. On the MKG-W dataset, PRISM reduces the average number of model invocations from 6 to 2.09 and lowers the configured inference cost by approximately 67.10%, while largely preserving the performance of the full-model ensemble. These results demonstrate that PRISM can effectively coordinate heterogeneous reasoning models while substantially reducing redundant computation. Our code and data are available at `https://anonymous.4open.science/r/PRISM-C687`

1 INTRODUCTION

Knowledge graphs (KGs) (Liang et al., 2024) represent entities and relations as structured triples and support applications such as natural language processing (Liu et al., 2025b) and recommender systems (Wang et al., 2019). However, KGs are inherently incomplete, motivating knowledge graph completion (KGC) to infer missing facts. Based on the information sources used, KGC methods can be broadly divided into unimodal KGC, which relies mainly on graph structure, and multimodal KGC, which additionally incorporates information such as text and images.

Unimodal KGC methods infer missing facts primarily from graph structure. Embedding-based methods (Bordes et al., 2013; Chao et al., 2021; Zhu et al., 2025b) model relational patterns in the embedding space, while graph neural network-based methods (Zhu et al., 2021) and rule-based methods (Tang et al., 2024) exploit neighborhood, path, and symbolic information. However, their effectiveness varies across queries: sparse graphs provide limited structural evidence, while incomplete rule coverage may leave some queries unsupported. Thus, a single reasoning mechanism may not adequately handle diverse query characteristics.

Beyond graph structure, entities are often associated with auxiliary information such as text and images. Multimodal KGC (Chen et al., 2023) incorporates such semantic information to improve missing-fact prediction. Representative methods, including MyGO (Zhang et al., 2025b), VISTA (Lee et al., 2023a), and MoMoK (Zhang et al., 2025a), mainly focus on multimodal repre-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

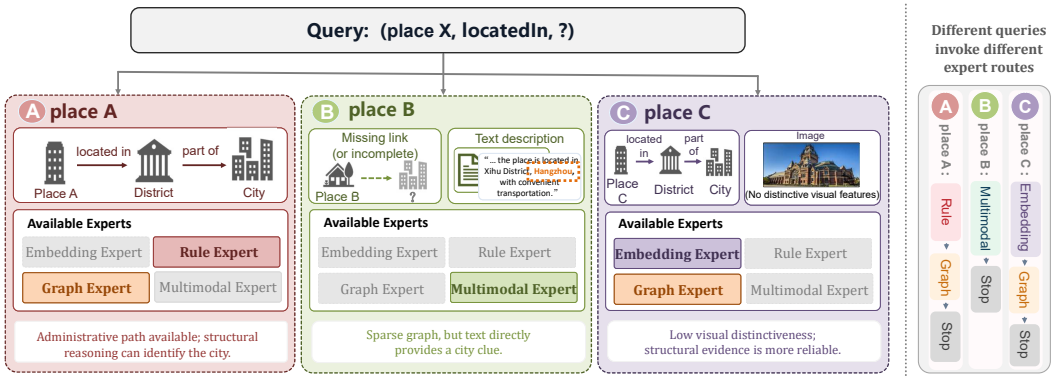

Figure 1: Queries with the same relation, `locatedIn`, may require substantially different reasoning information depending on the query entity.

sentation and fusion. However, modality information can be missing, noisy, or irrelevant to specific queries, where a query refers to an incomplete triple with a missing head or tail entity (Liu et al., 2025a; Peng et al., 2026). This limits the effectiveness of multimodal fusion. In contrast, graph- and rule-based reasoning can provide complementary structural evidence, suggesting that different information sources should be exploited adaptively for different queries.

As illustrated in Figure 1, consider the query $(placeX, \texttt{locatedIn}, ?)$. Although the relation is identical, different places may require different information. For *place A*, an administrative path from a district or county to its parent city can support graph-based or rule-based inference. For *place B*, whose neighborhood is sparse, a textual description explicitly mentioning the city gives multimodal models an advantage. For *place C*, an image of a visually indistinct building exterior may contribute little, while an available administrative path can make a structural expert more reliable. These cases show that selecting models solely by relation type cannot capture query-level differences in available information. This query-level heterogeneity calls for a unified mechanism that can dynamically assess and coordinate heterogeneous sources of information; however, existing work has yet to jointly integrate the structural reasoning strengths of unimodal KGC with the semantic modeling strengths of multimodal KGC in such a decision framework.

To this end, we propose PRISM, an agent framework that jointly coordinates unimodal KGC and multimodal KGC experts. PRISM encapsulates independently trained embedding, graph-based, rule-based, and multimodal models as callable experts and organizes their collaboration through a shared state. Its deliberation process consists of updating the reasoning state using query features, expert predictions, and structural support; its decision-making capability consists of selecting subsequent experts and determining when to stop under a budget constraint. Specifically, the PROFILER constructs relation and query profiles, the VERIFIER assesses structural support and prediction disagreement, the JUDGE fuses rankings, and the ROUTER selects experts. Together, these components form a traceable CALL–VERIFY–STOP loop that enables query-level complementarity between structural reasoning and multimodal modeling.

Our main contributions are as follows:

1. We analyze the limitations of fixed expert combinations in accommodating query differences and propose a sequential decision mechanism based on conditional error-correction gains. The mechanism dynamically updates the reasoning state using relation profiles, query information, and expert predictions, estimates the expected benefit of an additional expert, and autonomously selects subsequent experts or terminates inference under cost and budget constraints.
2. We introduce a KGC Agent framework that coordinates unimodal and multimodal completion models through expert selection, prediction updates, rank fusion, and budgeted stopping within a query-level loop.
3. Experiments on five unimodal and multimodal benchmarks evaluate predictive performance, expert invocation efficiency, and robustness. PRISM substantially reduces the num-

ber of expert calls and configuration costs while maintaining competitive link prediction performance.

2 Related Work

2.1 Unimodal Knowledge Graph Completion

Unimodal knowledge graph completion mainly exploits graph structure through embedding-based, rule-based, and graph neural network methods. Early approaches (Sun et al., 2019; Balažević et al., 2019) model entity–relation interactions through geometric or tensor operations. Recent methods further improve reasoning: FlorE (Duo et al., 2026) explores expressive geometric transformations, NBFNet (Zhu et al., 2021) performs relational path reasoning, and EpiGNN (Khalid & Schockaert, 2025) targets systematic relational reasoning. ULTRA (Galkin et al., 2024) focuses on transferable reasoning across knowledge graphs, while RNNLogic (Qu et al., 2020) and RulE (Tang et al., 2024) capture symbolic regularities. PRISM therefore treats heterogeneous KGC models as independent structural experts and selects them according to query context and intermediate predictions.

2.2 Multimodal Knowledge Graph Completion

Multimodal knowledge graph completion incorporates visual, textual, and other entity attributes to complement graph structure. Early methods (Xie et al., 2016; Xu et al., 2022) enhance structural representations with visual information. Recent approaches focus on cross-modal interaction and adaptive fusion. MKGFormer (Chen et al., 2022) uses multi-level visual–textual fusion, LAFA (Shang et al., 2024) reduces irrelevant visual information, while MyGO (Zhang et al., 2025b) and MoMoK (Zhang et al., 2025a) model fine-grained or relation-dependent multimodal features. However, their effectiveness can vary when modalities are missing, noisy, or irrelevant. PRISM instead treats independently trained unimodal KGC and multimodal KGC models as callable experts and dynamically selects complementary reasoning sources for each query.

3 Problem Formulation

3.1 Multimodal Knowledge Graph Completion

A multimodal knowledge graph is defined as

$$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{M}), \qquad \mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}, \tag{1}$$

where $\mathcal{E}$ and $\mathcal{R}$ denote the entity and relation sets, $\mathcal{T}$ is the observed triple set, and $\mathcal{M}$ contains multimodal information. For entity $e \in \mathcal{E}$, we denote its visual, textual, and structural information by $\mathcal{M}_e = (m_e^v, m_e^t, m_e^s)$. Given a tail-prediction query $q = (h, r, ?)$, an expert u assigns each candidate entity $x \in \mathcal{E}$ a score $s_u(q, x)$. Its average-tie rank is

$$\mathrm{rank}_u(q, x) = 1 + \big|x' : s_u(q, x') > s_u(q, x)\big| \quad + \frac{1}{2}\big|x' \neq x : s_u(q, x') = s_u(q, x)\big|. \tag{2}$$

Since score scales differ across experts, PRISM converts rankings into reciprocal rank fusion (RRF) scores (Cormack et al., 2009):

$$z_u(q, x) = \frac{1}{k_{\mathrm{rrf}} + \mathrm{rank}_u(q, x)}, \qquad k_{\mathrm{rrf}} > 0. \tag{3}$$

This transformation depends only on the ranking produced by each expert.

3.2 Budget-Constrained Sequential Expert Invocation

We treat each heterogeneous unimodal KGC or multimodal KGC model (e.g., InGram (Lee et al., 2023b), A*Net (Zhu et al., 2023), MCKGC (Gao et al., 2025), and APKGC (Jian et al., 2025)) as an expert. Each expert u has a positive invocation cost c_u. Under a total budget B and at most K expert calls, the routing policy Π_θ sequentially chooses either an untried, budget-feasible expert to Call, or Stop.

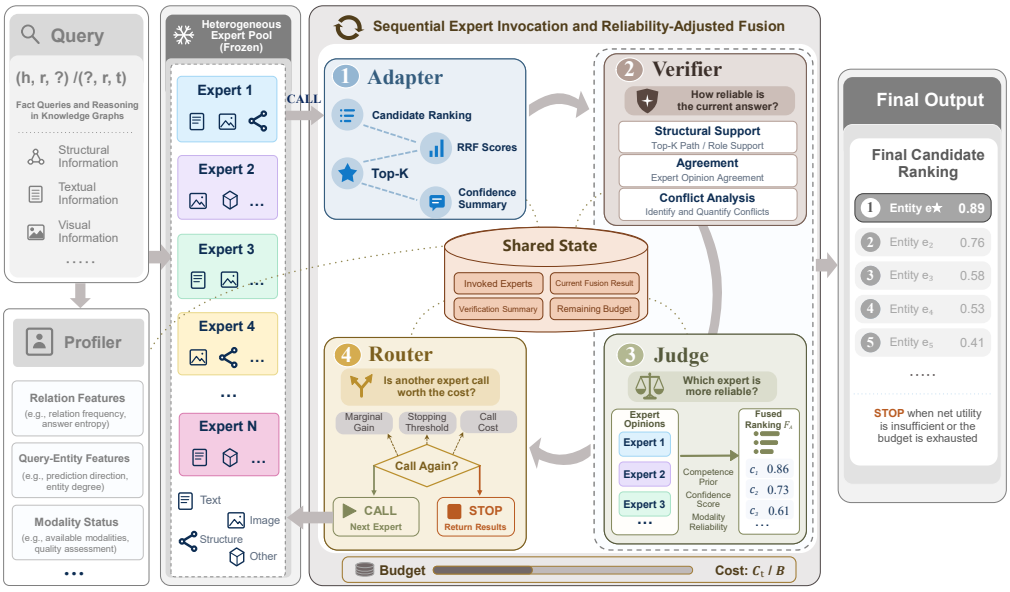

Figure 2: Overall framework of PRISM. Frozen experts are sequentially invoked under a budget, and their heterogeneous outputs are standardized into unified expert opinions. The VERIFIER and JUDGE then update the shared state for subsequent routing decisions.

Let D_t be the set of successfully invoked experts available by step t. The cumulative cost and remaining budget are

$$C_t = \sum_{u \in D_t} c_u, \qquad b_t = B - C_t, \qquad C_t \le B. \tag{4}$$

At termination step T_q, at least one expert must have succeeded, i.e., $D_{T_q} \neq \varnothing$. The JUDGE fuses the selected experts into scores $F_{D_{T_q}}(q, x)$. For the true target y_q, the reciprocal rank is

$$\mathrm{RR}(q; \Pi_\theta) = \frac{1}{\mathrm{rank}_{F_{D_{T_q}}}(q, y_q)}. \tag{5}$$

The ROUTER is trained to balance prediction quality and invocation cost:

$$\max_\theta \mathbb{E}_{q \sim \mathcal{D}} \left[\mathrm{RR}(q; \Pi_\theta) - \lambda C_{T_q} \right], \qquad \lambda \ge 0, \tag{6}$$

where $\mathcal{D}$ is the query set and λ controls the cost penalty.

4 METHODOLOGY

4.1 OVERALL FRAMEWORK AND INFERENCE PROCEDURE

As shown in Figure 2, PRISM coordinates frozen heterogeneous unimodal and multimodal KGC models through five components by maintaining a shared state: the PROFILER constructs query and expert priors, the ADAPTER standardizes heterogeneous expert outputs into unified expert opinions, the VERIFIER measures structural support and expert disagreement, the JUDGE fuses expert rankings, and the ROUTER selects CALL or STOP. At step t, PRISM maintains the shared state

$$S_t = \big(\phi(q), D_t, O_t, V_t, F_{D_t}, C_t, b_t\big), \tag{7}$$

where $\phi(q)$ is the query profile, O_t denotes expert opinions, V_t the verification statistics, and F_{D_t} the current fused ranking. After each successful call, PRISM updates the state and reassesses whether another expert should be invoked.

4.2 PROFILER: QUERY PROFILES AND EXPERT COMPETENCE

The PROFILER summarizes query characteristics from three aspects: relation features, query–entity features, and modality status. Relation features include relation frequency, answer entropy, and relation cardinality; query–entity features include prediction direction, entity degree, local connectivity, and answer multiplicity; and modality status describes the availability and quality of multimodal information. These features are directly computed and organized into the query profile $\phi(q)$, which is written into the shared state to guide the ROUTER and provide reliability priors for the JUDGE.

For each multimodal expert, we further estimate a modality reliability factor $\rho_u(q)$ that reflects its sensitivity to modality availability and quality by comparing the expert's MRR under complete multimodal inputs with that under modality-masked inputs.

4.3 ADAPTER: UNIFIED OPINIONS AND CONFIDENCE

The ADAPTER maps heterogeneous model outputs into a unified candidate space. An expert invocation returns

$$O_u(q) = \big(s_u, z_u, T_u^{K_{\text{top}}}, \eta_u\big), \tag{8}$$

where $s_u(q, x)$ is the output score of expert u, and $z_u(q, x)$ is the RRF score in Equation 3; $T_u^{K_{\text{top}}}(q)$ contains the top-K_{top} candidates ranked by z_u, while η_u summarizes output characteristics, including prediction uncertainty. We use a ridge regressor g_u^{conf} to estimate confidence from $\eta_u(q)$, and clip its output to $[0, 1]$:

$$\gamma_u(q) = \text{clip}\big(g_u^{\text{conf}}(\eta_u(q)), 0, 1\big). \tag{9}$$

The resulting confidence is used by both the ROUTER and the JUDGE.

4.4 VERIFIER: STRUCTURAL SUPPORT AND EXPERT AGREEMENT

The VERIFIER assesses the current prediction from two perspectives: structural support from the knowledge graph and agreement among invoked experts. Let $T_D^{K_{\text{top}}}(q)$ denote the Top-K_{top} candidates under the current fused ranking, $\mathcal{R}_{\text{side},r}$ denotes the set of entities observed in triples involving relation r, and $\mathcal{N}_2(q)$ the entities reachable from the query entity within two hops. We define the relation-role and path support as

$$\begin{aligned} \text{roleSup}_{K_{\text{top}}}(q, D) &= \frac{\left|T_D^{K_{\text{top}}}(q) \cap \mathcal{R}_{\text{side},r}\right|}{\left|T_D^{K_{\text{top}}}(q)\right|}, \\ \text{pathSup}_{K_{\text{top}}}(q, D) &= \frac{\left|T_D^{K_{\text{top}}}(q) \cap \mathcal{N}_2(q)\right|}{\left|T_D^{K_{\text{top}}}(q)\right|}. \end{aligned} \tag{10}$$

These quantities measure how much of the current candidate set is supported by relation statistics and local graph connectivity, respectively. The VERIFIER also measures agreement among the invoked experts. Let $x_u^{(1)}(q)$ denote the top-ranked candidate predicted by expert u. We define the top-1 agreement as

$$a_1(q, D) = \frac{1}{|D|} \max_x \sum_{u \in D} \mathbf{1}\left[x_u^{(1)}(q) = x\right], \tag{11}$$

which is the fraction of invoked experts that agree on the most frequently predicted top-ranked candidate. Let $T_u^{K_{\text{top}}}(q)$ denote the top-K_{top} candidate set of expert u. The mean pairwise Top-K_{top} overlap is defined as

$$\overline{o}_{K_{\text{top}}}(q, D) = \frac{2}{|D|(|D|-1)} \sum_{\substack{u,v \in D \\ u<v}} \frac{|T_u^{K_{\text{top}}}(q) \cap T_v^{K_{\text{top}}}(q)|}{K_{\text{top}}}, \tag{12}$$

which measures the average fraction of shared candidates between every pair of invoked experts. The conflict score is defined as

$$\text{conflict}(q, D) = \frac{1}{2}\left(1 - a_1(q, D)\right) + \frac{1}{2}\left(1 - \overline{o}_{K_{\text{top}}}(q, D)\right). \tag{13}$$

A larger conflict score indicates stronger disagreement among experts. These structural-support and agreement statistics are incorporated into the shared state and used by the ROUTER for subsequent CALL or STOP decisions.

4.5 JUDGE: RELIABILITY-ADJUSTED RANK FUSION

PRISM further estimates each expert's competence within a relation–direction cell using smoothed historical reciprocal rank. A relation–direction cell $\mathcal{Q}_{r,d}$ groups queries sharing the same relation r and prediction direction d, e.g., $\{(h_1, r, ?), (h_2, r, ?), \ldots\}$ for tail prediction. For expert u, let $R_{u,r,d}$ denote the accumulated reciprocal rank over $\mathcal{Q}_{r,d}$, and let $n_{r,d}$ denote the number of queries in that cell. Its competence is then estimated as

$$\pi_u(q) = \frac{R_{u,r,d} + \alpha_\pi \overline{\mathrm{RR}}_u}{n_{r,d} + \alpha_\pi}, \tag{14}$$

where $\overline{\mathrm{RR}}_u$ is the expert's historical reciprocal-rank prior and α_π controls the smoothing strength.

The JUDGE assigns each expert a quality score by combining modality reliability, competence, and confidence:

$$v_u(q) = \rho_u(q)\big((1-\beta)\pi_u(q) + \beta\gamma_u(q)\big)2^{-a_u(q)}, \tag{15}$$

where $a_u(q)$ applies soft abstention to low-confidence predictions, $\rho_u(q)$ is the modality reliability factor from Section 4.2, and $\gamma_u(q)$ is the prediction confidence defined in Equation 9. The quality score $v_u(q)$ is converted into a bounded fusion weight $w_u(q)$, and expert rankings are combined as

$$F_D(q, x) = \frac{\sum_{u \in D} w_u(q) z_u(q, x)}{\sum_{u \in D} w_u(q)}. \tag{16}$$

Rank-based normalization avoids incompatibility between heterogeneous raw-score scales, while reliability-aware weighting limits unreliable predictions. The fused result and verification statistics are then returned to the shared state for the next CALL–VERIFY–STOP decision.

4.6 ROUTER: UTILITY-BASED EXPERT SELECTION

The ROUTER estimates whether invoking an additional expert is beneficial given the current state. Given the shared state, the ROUTER computes a bounded risk intensity $\xi_D = \xi(q, D)$ by combining query-profile features (Section 4.2), prediction uncertainty (Section 4.3), and expert-agreement signals provided by the Verifier (Section 4.4). For expert u, the one-step gain from adding it to the current expert set D is defined as

$$y_u(q, D) = \mathcal{J}_{\xi_D}(q, D \cup \{u\}) - \mathcal{J}_{\xi_D}(q, D), \tag{17}$$

where $\mathcal{J}_{\xi_D}(q, D)$ is a risk-aware ranking reward dominated by reciprocal rank and augmented with small Top-10 and continuous-rank rewards in high-risk states. Subsequently, the modality reliability factor $\rho_u(q)$ is used to adjust the estimated gain of expert u. When the estimated gain is positive, reliability scaling is applied; when it is negative, its original value is retained. The resulting reliability-adjusted gain is denoted by $\widetilde{\Delta}_u(q, D)$. The ROUTER then assigns each feasible expert a routing utility score:

$$U_u(q, D) = \widetilde{\Delta}_u(q, D) - \lambda c_u + \kappa_u^{\mathrm{spec}} + \kappa^{\mathrm{cont}}, \tag{18}$$

where λ controls the penalty on expert-use cost. The terms κ_u^{spec} and κ^{cont} are the specialist and continuation biases, respectively, encouraging specialist invocation and additional information acquisition in difficult states. The ROUTER selects the feasible expert with the highest routing utility. After an initial expert is obtained, another expert is called only if its utility exceeds the continuation threshold; otherwise, the ROUTER selects STOP. Inference also terminates when the invocation limit or budget constraint is reached.

5 EXPERIMENTS

In this section, we introduce the basic experiment settings of our work and demonstrate our evaluation results with extensive analysis. The following six research questions (RQ) are the key questions that we explore in the experiments.

Table 1: Performance comparison on multimodal KGC benchmarks. Best results are shown in **bold**, and second-best results are underlined. YAGO15K results marked with † are taken from the comparative results reported by Moodle (Zhu et al., 2025a).

	MKG-W			MKG-Y			YAGO15K		
Model	**MRR (%)**	**H@1 (%)**	**H@10 (%)**	**MRR (%)**	**H@1 (%)**	**H@10 (%)**	**MRR (%)**	**H@1 (%)**	**H@10 (%)**
OTKGE (NeurIPS'22)	34.36	28.85	44.88	35.51	31.97	41.38	–	–	–
IMF (WWW'23)	34.50	28.77	45.44	35.79	32.95	42.63	34.50	27.60	49.00
QEB (ACM MM'23)	32.38	25.47	45.32	34.37	29.49	42.32	–	–	–
VISTA (EMNLP'23)	32.91	26.12	45.61	30.45	24.87	41.53	–	–	–
AdaMF-MAT (COLING'24)	35.85	29.04	48.42	38.57	34.34	45.76	34.40†	26.20†	48.60†
Moodle (NAACL'25)	–	–	–	–	–	–	37.70†	30.30†	51.80†
K-ON (AAAI'25)	35.83	29.41	47.16	35.83	32.56	42.45	–	–	–
MyGO (AAAI'25)	36.10	29.78	47.75	38.44	35.01	44.19	–	–	–
DVMKG (EMNLP'25)	–	–	–	–	–	–	41.90	**36.00**	53.40
MoMoK (ICLR'25)	35.89	30.38	46.13	37.91	35.09	43.20	–	–	–
All-expert equal RRF	38.55	32.16	**50.44**	**39.50**	35.54	**46.38**	25.67	16.94	42.95
PRISM	**38.57**	**32.67**	49.70	38.65	**35.64**	44.89	**41.91**	33.22	**55.46**

RQ1. Can PRISM achieve competitive performance on five benchmarks?

RQ2. Can PRISM reduce the number of expert calls and the overall inference cost?

RQ3. How much does each component and expert contribute to the final performance of PRISM?

RQ4. How robust is PRISM to missing modalities and variations in key parameters?

RQ5. Does PRISM provide advantages over simpler adaptive routing strategies?

RQ6. What do query groups and failure cases reveal about PRISM's routing and fusion?

5.1 Experimental Setup

Datasets. We evaluate PRISM on five KGC benchmarks. MKG-W (Xu et al., 2022), MKG-Y (Xu et al., 2022), and YAGO15K (Liu et al., 2019) are used for multimodal KGC, while WN18RR (Dettmers et al., 2018) and FB15K-237 (Toutanova & Chen, 2015) are used for unimodal KGC. Their statistics are summarized in Appendix Table C1.

Baseline Methods. We compare PRISM with representative multimodal and unimodal KGC methods. The **multimodal KGC baselines** include OTKGE (Cao et al., 2022), IMF (Li et al., 2023), QEB (Wang et al., 2023), VISTA (Lee et al., 2023a), AdaMF-MAT (Zhang et al., 2024b), K-ON (Guo et al., 2025), MyGO (Zhang et al., 2025b), MoMoK (Zhang et al., 2025a), Moodle (Zhu et al., 2025a) and DVMKG (Liu et al., 2025a) on MKG-W, MKG-Y, and YAGO15K. The **unimodal KGC baselines** include HAKE (Zhang et al., 2020), HousE (Li et al., 2022), CompoundE (Ge et al., 2023), SPAC (Yang et al., 2025), and KGCRR (Xu et al., 2025) on WN18RR and FB15K-237. PRISM uses six independently trained experts: PairRE, TuckER, NBFNet, RulE, MoMoK, and VISTA. For internal comparison, we consider the best single expert, relation-only expert selection, and equal-weight RRF over all six experts.

Metrics and settings. We report mean reciprocal rank (MRR), Hits@1, Hits@3, and Hits@10 under filtered evaluation (Bordes et al., 2013), excluding other known valid triples from the train, validation, and test splits. Complete results are provided in Appendix C2. Tied candidate scores use average ranks. PRISM permits at most $K = 3$ expert calls and uses cost penalty $\lambda = 0.01$.

5.2 Performance Comparison (RQ1)

Table 1 shows that PRISM is competitive across all three multimodal benchmarks. On MKG-W, it achieves the best MRR and Hits@1 (38.57%/32.67%), while all-expert RRF obtains the best Hits@10. On MKG-Y, PRISM obtains the best Hits@1 (35.64%), while all-expert RRF performs better in MRR and Hits@10. On YAGO15K, PRISM achieves the best MRR and Hits@10 (41.91%/55.46%), while DVMKG obtains the best Hits@1. These results indicate that adaptive

Table 2: Performance comparison on unimodal KGC benchmarks. Best results are shown in **bold**, and second-best results are underlined.

	WN18RR (%)			FB15K-237 (%)		
Model	**MRR**	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
HAKE (AAAI'20)	49.70	45.20	58.20	34.60	25.00	54.20
HousE (ICML'22)	51.10	46.50	60.20	36.10	26.60	55.10
CompoundE (ACL'23)	49.10	45.00	57.60	35.70	26.40	54.50
SPAC (AAAI'25)	50.10	46.20	59.10	37.20	28.00	56.20
KGCRR (AAAI'25)	48.40	44.60	55.20	42.80	33.70	60.20
PRISM	**55.26**	**50.02**	**65.91**	**42.85**	**34.67**	**60.48**

Table 3: Core ablations on MKG-W over three routing seeds (pp: percentage points).

Component	Setting	MRR (%, mean ± std.)	ΔMRR (pp)	Calls	Cost
Full PRISM		38.57 ± 0.07	–	2.09	1.84
Routing Design	(1.1) w/o context residual	38.18 ± 0.16	-0.40	1.97	1.69
	(1.2) w/o risk mechanism	38.43 ± 0.13	-0.14	1.81	1.59
	(1.3) w/o adaptive stopping	38.43 ± 0.07	-0.14	2.75	2.17
Fusion Design	(2.1) Equal-weight Judge	38.51 ± 0.06	-0.06	2.06	1.83

expert collaboration is effective, while the preferred expert combination varies across datasets and metrics. Results on the unimodal KGC benchmarks are reported in Table 2.

5.3 Performance–Cost Trade-off (RQ2)

PRISM achieves competitive ranking performance without exhaustive expert invocation. Across the three multimodal benchmarks, it uses only 1.71–2.09 experts per query and reduces configured cost by 67.10%–74.60%. Additional calls can be beneficial, redundant, or detrimental depending on the dataset, supporting selective rather than exhaustive invocation; the complete cross-dataset results and analysis are provided in Appendix Figure 5 and Table C2.

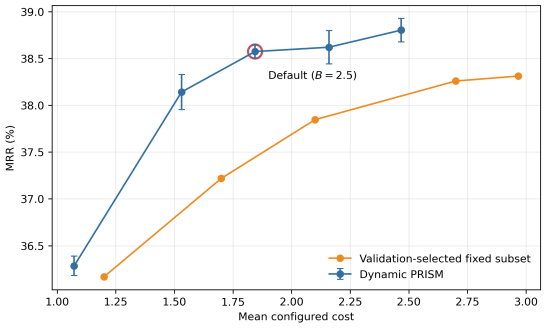

Figure 3: Budget–performance trade-off on MKG-W ($K = 3$). Dynamic results show the mean and standard deviation over three routing seeds. A fixed subset is selected on the validation utility-fit partition at each budget and then used for every test query with equal-weight RRF.

Figure 3 further compares dynamic routing with a fixed-subset strategy under the same budget and three-call limit. The fixed baseline selects one expert subset on the validation utility-fit partition and applies it to all test queries, whereas PRISM adapts the invoked experts to each query. Dynamic routing achieves higher MRR at every tested budget. At the default budget $B = 2.50$, PRISM obtains 38.57% MRR with a mean configured cost of 1.84, compared with 37.85% MRR and a cost of 2.10 for the fixed subset. Increasing the budget from 2.50 to 3.00 yields only about 0.05% additional MRR while increasing mean cost by about 0.32, suggesting diminishing returns from further expert calls.

5.4 Ablation Study (RQ3)

Table 3 validates the main design choices of PRISM. Removing the context residual yields the largest MRR drop of 0.39 percentage points. Removing the risk mechanism reduces the config-

Table 4: Comparison with adaptive routing baselines on MKG-W.

Model	MRR (%)	H@1 (%)	H@3 (%)	H@10 (%)	Calls	Cost
LearnedRouter-K	37.42	30.82	39.80	49.34	2.04	1.73
ConfidenceCascade	36.60	30.36	39.06	48.32	**1.44**	**0.74**
PRISM	**38.57**	**32.67**	**40.59**	**49.70**	2.09	1.84

ured cost from 1.84 to 1.59, with an MRR loss of 0.14 percentage points, while disabling adaptive stopping increases the average number of expert calls from 2.09 to 2.75 without improving MRR. Equal-weight fusion also slightly degrades MRR, supporting the benefit of reliability-aware fusion. Overall, the results show that the routing components mainly govern the performance–cost trade-off, while the JUDGE provides an additional refinement to the final ranking.

5.5 MISSING MODALITIES AND HYPERPARAMETER SENSITIVITY (RQ4)

We evaluate PRISM's robustness to missing modalities. On MKG-W, we randomly mask visual or textual features for nested sets of 10%, 30%, and 50% of entities using seed 2026. Compared with strong multimodal KGC baselines, including NativE (Zhang et al., 2024a), APKGC (Jian et al., 2025), and M-Hyper (Liu et al., 2026), PRISM achieves the highest average MRR across corruption levels, demonstrating stronger overall robustness. Its MRR remains nearly unchanged under 50% visual missingness (38.57%→38.40%) and competitive under severe textual missingness.

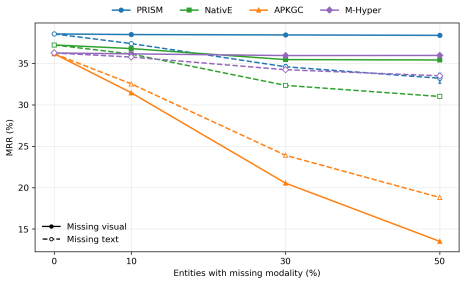

Figure 4: Missing-modality robustness.

5.6 COMPARISON WITH ADAPTIVE ROUTING BASELINES (RQ5)

We compare PRISM with two adapted routing baselines on MKG-W: LearnedRouter-K, inspired by RouteLLM (Ong et al., 2025), and ConfidenceCascade, inspired by AutoMix (Aggarwal et al., 2024). LearnedRouter-K performs one-shot query-conditioned expert selection, whereas ConfidenceCascade uses a fixed expert order with adaptive stopping. As shown in Table 4, PRISM achieves 38.57% MRR, outperforming LearnedRouter-K (37.42%) and ConfidenceCascade (36.60%) by 1.13 and 1.95 percentage points, respectively. LearnedRouter-K attains a similar cost to PRISM but lower accuracy, whereas ConfidenceCascade further reduces cost at a larger performance loss. These results indicate that both sequential state updates and query-dependent expert selection contribute to PRISM's performance.

5.7 QUERY GROUPS AND FAILURE ANALYSIS (RQ6)

Head prediction is substantially more challenging than tail prediction on MKG-W, with lower MRR (20.15% vs. 57.00%) and more expert calls (2.39 vs. 1.80). RulE, NBFNet, and VISTA provide the largest marginal gains, highlighting the benefit of complementary structural and multimodal evidence. Additional expert calls can also be detrimental when predictions are weak or correlated, as reflected by the large gap between all-expert equal RRF and PRISM on YAGO15K (25.67% vs. 41.91% MRR). Detailed analyses are provided in Appendix Table C9.

6 CONCLUSION

We presented PRISM, a query-adaptive framework that coordinates heterogeneous KGC and MMKGC models as callable experts. By combining utility-based routing, verification, and reliability-aware fusion, PRISM balances prediction quality and inference cost, reducing configured cost by 67.10%–74.60% on three multimodal benchmarks. Future work will focus on better stopping calibration and dependency-aware expert fusion.

AI USE STATEMENT

We used an AI assistant to translate Chinese manuscripts into English, polish and refine the language, assist with LaTeX formatting, and support data processing and the implementation of experimental details.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. Automix: Automatically mixing language models. *Advances in Neural Information Processing Systems*, 37:131000–131034, 2024.

Ivana Balažević, Carl Allen, and Timothy Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 5185–5194, 2019.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. Otkge: Multi-modal knowledge graph embeddings via optimal transport. *Advances in neural information processing systems*, 35:39090–39102, 2022.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. Pairre: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 4360–4369, 2021.

Xiang Chen, Ningyu Zhang, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, and Huajun Chen. Hybrid transformer with multi-level fusion for multimodal knowledge graph completion. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 904–915, 2022.

Zhuo Chen, Jiaoyan Chen, Wen Zhang, Lingbing Guo, Yin Fang, Yufeng Huang, Yichi Zhang, Yuxia Geng, Jeff Z Pan, Wenting Song, et al. Meaformer: Multi-modal entity alignment transformer for meta modality hybrid. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 3317–3327, 2023.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zehua Duo, Jiang Li, Xiangdong Su, and Guanglai Gao. Flore: Integrating full lorentz group and directional offsets for effective knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20968–20976, 2026.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 31598–31619, 2024.

Yuxiao Gao, Fuwei Zhang, Zhao Zhang, Xiaoshuang Min, and Fuzhen Zhuang. Mixed-curvature multi-modal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11699–11707, 2025.

Xiou Ge, Yun Cheng Wang, Bin Wang, and C-C Jay Kuo. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 6947–6965, 2023.

Lingbing Guo, Yichi Zhang, Zhongpu Bo, Zhuo Chen, Mengshu Sun, Zhiqiang Zhang, Wen Zhang, and Huajun Chen. K-on: Stacking knowledge on the head layer of large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11745–11753, 2025.

Yue Jian, Xiangyu Luo, Zhifei Li, Miao Zhang, Yan Zhang, Kui Xiao, and Xiaoju Hou. Apkgc: Noise-enhanced multi-modal knowledge graph completion with attention penalty. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 15005–15013, 2025.

Irtaza Khalid and Steven Schockaert. Systematic relational reasoning with epistemic graph neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 41022–41060, 2025.

Jaejun Lee, Chanyoung Chung, Hochang Lee, Sungho Jo, and Joyce Jiyoung Whang. Vista: Visual-textual knowledge graph representation learning. In *Findings of the association for computational linguistics: EMNLP 2023*, pp. 7314–7328, 2023a.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. Ingram: Inductive knowledge graph embedding via relation graphs. In *International conference on machine learning*, pp. 18796–18809. PMLR, 2023b.

Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, et al. House: Knowledge graph embedding with householder parameterization. In *International conference on machine learning*, pp. 13209–13224. PMLR, 2022.

Xinhang Li, Xiangyu Zhao, Jiaxing Xu, Yong Zhang, and Chunxiao Xing. Imf: interactive multimodal fusion model for link prediction. In *Proceedings of the ACM web conference 2023*, pp. 2572–2580, 2023.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, Fuchun Sun, and Kunlun He. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal. *IEEE transactions on pattern analysis and machine intelligence*, 46 (12):9456–9478, 2024.

Minghang Liu, Yinghan Shen, Zihe Huang, Yuanzhuo Wang, Xuhui Jiang, and Huawei Shen. Differentiated vision: Unveiling entity-specific visual modality requirements for multimodal knowledge graph. In *EMNLP (Findings)*, pp. 20170–20183, 2025a.

Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S Rosenblum. Mmkg: multi-modal knowledge graphs. In *European Semantic Web Conference*, pp. 459–474. Springer, 2019.

Zhiqiang Liu, Chengtao Gan, Junjie Wang, Yichi Zhang, Zhongpu Bo, Mengshu Sun, Huajun Chen, and Wen Zhang. Ontotune: Ontology-driven self-training for aligning large language models. In *Proceedings of the ACM on Web Conference 2025*, pp. 119–133, 2025b.

Zhiqiang Liu, Yichi Zhang, Mengshu Sun, Lei Liang, and Wen Zhang. Collaboration of fusion and independence: Hypercomplex-driven robust multi-modal knowledge graph completion. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27968–27983, 2026.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Jiashun Peng, Fu Zhang, Hongzhi Chen, Jingwei Cheng, Yingsong Ning, and Xiaoke Wang. Not all modalities at once: Dynamic dropout and bidirectional fusion for robust multi-modal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17928–17940, 2026.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *arXiv preprint arXiv:2010.04029*, 2020.

Bin Shang, Yinliang Zhao, Jun Liu, and Di Wang. Lafa: Multimodal knowledge graph completion with link aware fusion and aggregation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 8957–8965, 2024.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Xiaojuan Tang, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. Rule: Knowledge graph reasoning with rule embedding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4316–4335, 2024.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pp. 57–66, 2015.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 950–958, 2019.

Xin Wang, Benyuan Meng, Hong Chen, Yuan Meng, Ke Lv, and Wenwu Zhu. Tiva-kg: A multimodal knowledge graph with text, image, video and audio. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 2391–2399, 2023.

Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Image-embodied knowledge representation learning. *arXiv preprint arXiv:1609.07028*, 2016.

Derong Xu, Tong Xu, Shiwei Wu, Jingbo Zhou, and Enhong Chen. Relation-enhanced negative sampling for multimodal knowledge graph completion. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 3857–3866, 2022.

Kuan Xu, Kuo Yang, Jian Liu, Xiangkui Lu, Jun Wu, and Xuezhong Zhou. Kgcrr: An effective metric-driven knowledge graph completion framework by designing a novel upper bound function with adaptive approximation to reciprocal rank. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12927–12935, 2025.

Chuhong Yang, Bin Li, and Nan Wu. Spac: Sparse partitioning and adaptive core tensor pruning model for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 15230–15238, 2025.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Binbin Hu, Ziqi Liu, Wen Zhang, and Huajun Chen. Native: Multi-modal knowledge graph completion in the wild. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 91–101, 2024a.

Yichi Zhang, Zhuo Chen, Lei Liang, Huajun Chen, and Wen Zhang. Unleashing the power of imbalanced modality information for multi-modal knowledge graph completion. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 17120–17130, 2024b.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Binbin Hu, Ziqi Liu, Wen Zhang, Huajun Chen, et al. Multiple heads are better than one: Mixture of modality knowledge experts for entity representation learning. In *International Conference on Learning Representations*, volume 2025, pp. 54811–54828, 2025a.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Binbin Hu, Ziqi Liu, Wen Zhang, and Huajun Chen. Tokenization, fusion, and augmentation: towards fine-grained multi-modal entity representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13322–13330, 2025b.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3065–3072, 2020.

Guoliang Zhu, Tao Ren, Dandan Wang, and Jun Hu. Let modalities teach each other: modal-collaborative knowledge extraction and fusion for multimodal knowledge graph completion. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6210–6222, 2025a.

Yushan Zhu, Wen Zhang, Zhiqiang Liu, Mingyang Chen, Lei Liang, and Huajun Chen. Croppable knowledge graph embedding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11818–11835, 2025b.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in neural information processing systems*, 34:29476–29490, 2021.

Zhaocheng Zhu, Xinyu Yuan, Michael Galkin, Louis-Pascal Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. A* net: A scalable path-based reasoning approach for knowledge graphs. *Advances in neural information processing systems*, 36:59323–59336, 2023.

A NOTATION

Table A1: Notation used throughout the paper.

Symbol	Meaning
$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{M})$	Knowledge graph with entity, relation, triple, and multimodal-information sets.
$\mathcal{T}_{\mathrm{tr}}, \mathcal{T}_{\mathrm{val}}$	Training and validation triples.
q, x, y_q	Query, candidate entity, and true target used only for offline supervision or evaluation.
$\mathcal{X}, u$	Expert pool and an expert.
$s_u, \mathrm{rank}_u, z_u, k_{\mathrm{rrf}}$	Raw expert score, average-tie rank, RRF score, and RRF constant.
K, K_{top}	Maximum number of expert calls and candidate-list cutoff.
D_t, O_t, V_t	Successfully invoked experts, their opinions, and verification statistics at step t.
$c_u, C_t, b_t, B, \Pi_\theta$	Expert cost, spent cost, remaining budget, total budget, and routing policy.
$P, \phi(q), S_t$	Training-graph relation profile, query profile, and shared state.
$O_u(q), \eta_u(q)$	Standardized expert opinion and its output-summary features.
$\pi_u(q), \gamma_u(q), \rho_u(q)$	Relation–direction competence prior, confidence estimate, and modality reliability factor.
$\xi_D = \xi(q, D)$	Bounded risk intensity for the current expert set.
$\mathcal{J}_{\xi_D}(q, D), y_u(q, D)$	Risk-aware ranking reward and offline gain label.
$\widetilde{\Delta}_u(q, D)$	Reliability-adjusted gain.
$U_u(q, D)$	Routing utility for a feasible expert.
$a_u(q), v_u(q), w_u(q)$	Soft-abstention indicator, Judge quality score, and fusion weight.
$F_D(q, x)$	Weighted RRF score for the successfully invoked expert set D.
$\mathcal{C}, \mathcal{U}, \mathcal{H}$	Disjoint validation partitions for competence estimation, utility fitting, and policy acceptance.

Table A1 summarizes the symbols used by the routing, verification, and fusion components. In particular, D_t consistently denotes the experts available at step t, while K and K_{top} distinguish the expert-call limit from the candidate-list cutoff.

B ROUTER AND JUDGE HYPERPARAMETERS

Table B1 lists the default Router and Judge configuration used in the reported experiments unless a sensitivity setting explicitly overrides one factor. The separate values of $K = 3$ and $K_{\mathrm{top}} = 10$ control invocation depth and candidate-level verification, respectively.

Table B1: Default Router and Judge configuration.

Parameter	Value	Meaning
Budget B / call limit K	2.50 / 3	Internal budget for expert invocation.
Continuation threshold / cost penalty λ	0 / 0.01	Stopping threshold and penalty on expert cost.
RRF constant k_{rrf} / ridge penalty	10 / 10	Rank-fusion and regression regularization.
Maximum utility subsets	64	Enumeration limit; 22 subsets used.
Path-edge budget / relation samples	256 / 32	Verification search limits.
Residual minimum / shrinkage / clipping	8 / 20 / 0.05	Context-residual settings.
risk threshold	0.50	Gate for runtime risk bonuses.
head / N–N / high-degree	0.35 / 0.35 / 0.30	Structural-risk weights.
margin / entropy / agreement	0.05 / 0.90 / 0.75	Confidence-risk thresholds.
Specialist / continuation biases	0.04 / 0.02	κ_u^{spec} and κ^{cont}.
Judge strength / blend β	0.50 / 0.50	Weight mapping and prior–confidence blend.
Judge weight range	[0.25, 4]	Weight clipping interval.
abstain threshold	0.02	Halve quality without removing the expert.
Judge K_{top}	10	Overlap and opinion summaries.

B.1 Budget and Expert-Cost Configuration

PRISM uses a normalized budget of $B = 2.50$. The configured expert costs are 0.20 (PairRE), 0.40 (TuckER), 1.00 (MoMoK), 1.00 (RulE), 1.50 (NBFNet), and 1.50 (VISTA), summing to 5.60 for invoking all six experts. These values are architecture-based proxy costs rather than measurements of wall-clock latency. They define three coarse complexity tiers: PairRE and TuckER are lightweight embedding models; RulE and MoMoK require rule-based reasoning or multimodal fusion; and NBFNet and VISTA involve graph message passing or heavier multimodal inference. The budget therefore permits several complementary low- or medium-cost calls, or one high-cost expert together with lighter experts, while preventing exhaustive invocation and creating a meaningful routing trade-off. We use these proxy values only as the Router's internal accounting mechanism; the practical efficiency claim is independently supported by the synchronized end-to-end latency and throughput measurements in Appendix Section C.3, rather than inferred from the proxy costs alone.

C Supplementary Results

C.1 Dataset Statistics

Table C1: Dataset statistics.

Dataset	Entities	Relations	Train	Valid	Test
MKG-W	15,000	169	34,196	4,276	4,274
MKG-Y	15,000	28	21,310	2,665	2,663
YAGO15K	15,404	32	86,020	12,289	24,577
WN18RR	40,943	11	86,835	3,034	3,134
FB15K-237	14,541	237	272,115	17,535	20,466

Table C1 summarizes the scale and split sizes of all five benchmarks. The three multimodal datasets have similar entity counts but substantially different relation and training densities, while WN18RR and FB15K-237 provide complementary unimodal settings with markedly different numbers of entities and relations.

C.2 Performance–Cost Trade-offs

Figure 5 places MRR alongside configured cost and shows that invoking more experts does not necessarily improve ranking performance. On MKG-W, PRISM achieves nearly the same MRR as all-expert RRF at substantially lower configured cost. On MKG-Y, all-expert RRF improves MRR by only 0.85 percentage points, whereas on YAGO15K it performs substantially worse than PRISM; relation-only selection slightly exceeds PRISM in MRR. Thus, additional experts can be beneficial, redundant, or even detrimental depending on the query distribution, motivating selective rather than exhaustive expert invocation.

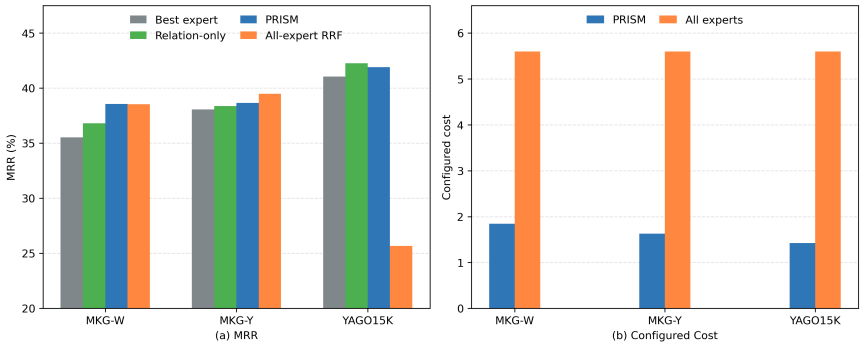

Figure 5: MRR and configured cost under a unified evaluation protocol on the three multimodal datasets.

Table C2: PRISM ranking metrics, mean calls, and mean configured cost.

Dataset	MRR (%)	H@1 (%)	H@3 (%)	H@10 (%)	Calls	Cost	Cost reduction
MKG-W	38.57	32.67	40.59	49.70	2.09	1.84	67.10%
MKG-Y	38.65	35.64	40.20	44.89	2.01	1.63	70.90%
YAGO15K	41.91	33.22	44.66	55.46	1.71	1.42	74.60%
WN18RR	55.26	50.02	57.32	65.91	1.97	1.69	45.40%
FB15K-237	42.85	34.67	46.05	60.48	2.19	1.77	15.70%

Table C2 summarizes the resulting cost across datasets. On the three multimodal benchmarks, PRISM invokes only 1.71–2.09 experts per query on average, reducing expert calls by 65.10%–71.60% and configured cost by 67.10%–74.60% relative to invoking all six experts. The mean configured costs are 1.84 on MKG-W, 1.63 on MKG-Y, and 1.42 on YAGO15K, corresponding to cost reductions of 67.10%, 70.90%, and 74.60%, respectively. On WN18RR, PRISM uses 1.97 experts per query with a mean cost of 1.69, still reducing configured cost by 45.40%. Overall, PRISM achieves the performance–cost trade-off primarily by avoiding unnecessary expert evaluations and adapting expert use to each query.

C.3 Measured End-to-End Inference Efficiency

Because the configured costs in Table C2 are routing proxies rather than hardware measurements, we additionally benchmark end-to-end inference under a controlled protocol. All methods use the same six resident experts, the same ordered set of 8,548 bidirectional MKG-W test queries, and batch size one on a single NVIDIA GeForce RTX 3090. Each query is timed with CUDA synchronization immediately before and after inference. Model loading, warm-up, file I/O, provenance checks, and gold-dependent filtered-rank computation are excluded, whereas query profiling, routing, expert scoring, verification, and fusion required by each method are included. Each method is run in a fresh process for three full timing repetitions after 32 warm-up queries.

Table C3 confirms that PRISM's efficiency advantage is not merely an artifact of the configured cost units. Relative to all-expert equal RRF, PRISM reduces the mean number of expert invocations by 65.10% while the absolute MRR difference remains below 0.01 percentage points. More importantly, it reduces median latency by $17.80\times$, reduces P95 latency by $4.50\times$, and increases sequential throughput by $11.50\times$. Compared with the all-expert Judge, PRISM remains substantially faster while retaining competitive ranking quality. These results demonstrate the practical benefit of query-adaptive routing: PRISM preserves almost all of the accuracy of exhaustive fusion while avoiding most redundant expert computation.

The full-pool-resident protocol deliberately gives all methods access to the same already-loaded models and therefore isolates per-query execution rather than model-loading effects. Under this conservative setup, PRISM does not claim a clear peak-memory advantage, because all six models remain resident. Its primary measured benefits are lower latency and higher throughput. PRISM's per-repetition throughput is 8.17, 18.64, and 14.96 queries/s, whereas both exhaustive baselines are

Table C3: Measured end-to-end inference efficiency on MKG-W. Latencies are milliseconds per query; throughput is sequential queries per second. Peak memory reports PyTorch CUDA allocated/reserved GiB.

Method	MRR (%)	Calls	P50 (ms)	P95 (ms)	Query/s	Peak GiB (A/R)
PRISM	38.57	2.09	54.90	257.63	12.35	1.62/3.34
All-expert equal RRF	38.55	6.00	979.55	1157.51	1.08	1.76/2.26
All-expert Judge	38.68	6.00	998.21	1162.32	1.05	1.76/2.26

stable near 1.05–1.08 queries/s. We therefore report the complete latency distribution and repetitions rather than relying only on a single speedup number; nevertheless, every repetition retains a large efficiency advantage over exhaustive invocation.

C.4 Unified Evaluation of the Six Experts

Table C4: Complete filtered ranking metrics for the six experts on the three multimodal datasets.

Dataset	Expert	MRR (%)	H@1 (%)	H@3 (%)	H@10 (%)
MKG-W	RulE	35.53	29.97	37.53	46.02
	MoMoK	35.08	29.75	36.98	45.29
	VISTA	35.02	28.09	37.92	48.00
	PairRE	34.97	29.29	37.34	45.62
	TuckER	32.61	27.23	35.07	43.18
	NBFNet	31.85	24.06	34.90	45.92
MKG-Y	RulE	38.07	34.87	39.50	43.80
	MoMoK	37.75	34.85	39.24	42.60
	TuckER	37.05	34.53	38.32	41.64
	PairRE	36.83	32.73	39.20	43.47
	NBFNet	31.26	24.13	35.34	43.71
	VISTA	30.23	24.35	32.65	41.55
YAGO15K	RulE	41.05	34.43	44.12	53.39
	TuckER	32.91	25.52	36.10	46.95
	VISTA	23.64	15.71	25.83	39.09
	MoMoK	22.89	15.61	24.84	37.13
	NBFNet	17.85	12.00	18.86	28.96
	PairRE	15.73	9.50	16.99	27.95

Table C4 provides the complete filtered results of all six experts. RulE is the strongest individual expert in MRR on each multimodal benchmark, but the performance gaps and ranking patterns vary considerably across datasets, supporting the need for query-dependent expert selection rather than a single fixed expert.

C.5 Complete Component Ablations

Table C5: Component ablations on MKG-W (mean over three routing seeds); pp denotes percentage points.

ID	Setting	MRR (%, mean $\pm$ std.)	ΔMRR (pp)	Calls	Cost
A0	Full PRISM	38.57 ± 0.07	$+0.00$	2.09	1.84
A1	Without local query structure	38.48 ± 0.17	-0.09	2.00	1.79
A2	Global competence prior	38.47 ± 0.06	-0.10	2.11	1.86
A3	Without context residual	38.18 ± 0.16	-0.40	1.96	1.69
A4	Without risk mechanism	38.43 ± 0.13	-0.14	1.81	1.59
A5	Without adaptive stopping	38.43 ± 0.07	-0.14	2.75	2.17
A6	Without disagreement feedback	38.55 ± 0.14	-0.02	2.05	1.85
A7	Equal-weight Judge	38.51 ± 0.06	-0.06	2.06	1.83

Table C5 reports the contribution of each routing and fusion component. Removing the context residual causes the largest MRR decrease among the individual component ablations, whereas disabling adaptive stopping increases the mean number of calls from 2.09 to 2.75. The smaller changes for several other components indicate that their effects are complementary rather than uniformly additive.

C.6 Expert-Removal Ablations

Table C6: Expert-removal ablations on MKG-W (mean over three routing seeds); pp denotes percentage points.

ID	Removed expert	MRR (%, mean $\pm$ std.)	ΔMRR (pp)	Calls	Cost
A0	None	38.57 ± 0.07	–	2.09	1.84
E1	PairRE	38.55 ± 0.13	-0.02	1.81	2.02
E2	TuckER	38.40 ± 0.07	-0.18	1.82	1.84
E3	NBFNet	38.65 ± 0.07	$+0.07$	2.05	1.82
E4	RulE	38.38 ± 0.12	-0.20	2.19	1.71
E5	MoMoK	38.51 ± 0.05	-0.06	2.14	1.74
E6	VISTA	37.12 ± 0.08	-1.45	2.18	1.72

Table C6 measures how the expert pool changes when one expert is removed. Excluding VISTA produces the largest MRR decrease, while removing NBFNet yields a small increase, showing that an expert's standalone MRR does not directly determine its value within the routed ensemble.

C.7 Missing Modalities and Hyperparameters

Table C7: Missing-modality robustness on MKG-W (three-seed means; reliability adjustment enabled); pp denotes percentage points.

Missing modality	Rate	MRR (%)	ΔMRR from A0 (pp)	Calls	Cost
None	0%	38.57	–	2.09	1.84
Visual	10%	38.50	-0.08	2.10	1.84
Visual	30%	38.45	-0.12	2.09	1.84
Visual	50%	38.40	-0.17	2.09	1.84
Textual	10%	37.41	-1.17	2.10	1.85
Textual	30%	34.61	-3.97	2.09	1.85
Textual	50%	33.22	-5.36	2.10	1.85

Table C7 shows that performance declines smoothly as modality information is removed. Textual missingness has a substantially larger effect than visual missingness on MKG-W, while the mean number of calls and configured cost remain nearly unchanged across the tested rates.

Table C8 reports one-factor sensitivity around the default configuration. The tested changes produce relatively small MRR variations, although stronger cost penalization reduces both configured cost and MRR, illustrating the expected performance–efficiency trade-off.

C.8 Query Groups and Marginal Expert Gains

The query-group results complement the aggregate efficiency analysis by showing where adaptive expert allocation is most useful. Head-prediction and high-degree queries are substantially harder and trigger more expert calls, while NBFNet, RulE, and VISTA provide the largest average marginal gains. This pattern supports PRISM's central advantage: it allocates additional computation to difficult queries and avoids exhaustive invocation when further experts are unlikely to help.

Table C8: One-factor hyperparameter sensitivity on MKG-W (three-seed means); pp denotes percentage points.

ID	Setting	MRR (%)	ΔMRR from A0 (pp)	Calls	Cost
A0	Default	38.57	–	2.09	1.84
C0	Cost penalty $= 0$	38.61	$+0.04$	2.04	2.07
C2	Cost penalty $= 0.02$	38.36	-0.22	2.07	1.62
J25	Judge strength $= 0.25$	38.63	$+0.06$	2.10	1.84
J75	Judge strength $= 0.75$	38.60	$+0.02$	2.08	1.85
R35	Risk threshold $= 0.35$	38.63	$+0.06$	2.18	1.94
R65	Risk threshold $= 0.65$	38.43	-0.14	1.96	1.72

Table C9: Query groups and marginal gains from additional experts on MKG-W. ΔRR is the mean change in reciprocal rank after adding an expert.

Category	Query group or expert	MRR / ΔRR (%)	Calls / positive-gain rate
Query direction	Head prediction	20.15	2.39 calls
Query direction	Tail prediction	57.00	1.80 calls
Structural difficulty	High-degree query	10.26	2.67 calls
Structural difficulty	N–N relation	18.44	–
Additional expert	NBFNet	+2.48	65.50% positive
Additional expert	RulE	+2.58	–
Additional expert	VISTA	+1.71	56.50% positive
Additional expert	PairRE / TuckER	+0.57 / +0.22	Invoked frequently

D Implementation Details

D.1 Training and Inference Algorithms

Routing-policy training. Experts are trained independently and then frozen. The Router uses only the training graph and validation set: the former supplies relation profiles, and validation triples are divided by triple into competence fitting, utility fitting, and holdout acceptance using seed 42. The two prediction directions of a triple remain in the same partition, preventing direction-level leakage.

Algorithm 1: Routing-policy fitting with disjoint validation partitions. **Input:** Training triples $\mathcal{T}_{\text{tr}}$, validation triples $\mathcal{T}_{\text{val}}$, frozen experts $\mathcal{X}$, seed 42.
Output: Relation profile P and routing policy Π_θ.

1. Set $P \leftarrow \text{ProfileRelations}(\mathcal{T}_{\text{tr}})$.
2. Split $\mathcal{T}_{\text{val}}$ into disjoint $\mathcal{C}, \mathcal{U}, \mathcal{H}$ partitions (20%, 60%, 20%) using seed 42.
3. For each $u \in \mathcal{X}$, estimate the relation–direction prior π_u, confidence model g_u^{conf}, and reliability fingerprint on $\mathcal{C}$.
4. For each query q and invoked set D sampled on $\mathcal{U}$, compute the gain label $y_u(q, D)$ from adding each feasible expert.
5. Fit the gain estimator and context residuals using states without access to the true target.
6. Store the fitted competence priors, confidence and gain estimators, and context residuals in Π_θ.
7. Accept the candidate policy only on $\mathcal{H}$ and then freeze it.

True validation targets are used only to construct supervision on $\mathcal{C}$ and $\mathcal{U}$; $\mathcal{H}$ does not fit parameters.

Algorithm 2: Budgeted expert invocation and stopping. **Input:** Query q, relation profile P, frozen experts $\mathcal{X}$, policy Π_θ, budget B, and maximum number of calls K.
Output: A full-entity fused ranking and execution trace.

1. Initialize $t \leftarrow 0$, invoked experts $D_0 \leftarrow \varnothing$, spent cost $C_0 \leftarrow 0$, remaining budget $b_0 \leftarrow B$, and state $S_0 \leftarrow \mathrm{ProfileQuery}(q, P)$.
2. While $|D_t| < K$, form the untried feasible set $\{u \in \mathcal{X} \setminus D_t : c_u \leq b_t\}$; STOP if it is empty.
3. Compute $U_u(q, D_t)$ for feasible experts and select u^* deterministically by decreasing utility, then lower cost, then expert name.
4. If $D_t \neq \varnothing$ and U_{u^*} does not exceed the continuation threshold, STOP.
5. Invoke u^*, standardize its output as $O_{u^*}(q)$, and set $D_{t+1} \leftarrow D_t \cup \{u^*\}$, $C_{t+1} \leftarrow C_t + c_{u^*}$, and $b_{t+1} \leftarrow B - C_{t+1}$.
6. The Verifier computes soft training-graph support; opinion summaries capture agreement and conflict.
7. The Judge updates expert weights and $F_{D_{t+1}}$; refresh S_{t+1}, increment t, and repeat.
8. If $D_t = \varnothing$, return an error. Otherwise, output the average-tie ranking and full trace.

D.2 Complexity

The relation profile requires one pass over the training triples, or $O(|\mathcal{T}_{\mathrm{tr}}|)$ time. Path checks are bounded by 32 positive examples per relation and a 256-edge budget per query. Let N be the number of entities, $m = |D| \leq K$ the number of invoked experts, and c_u the configured cost of expert u. The current implementation has per-query time complexity

$$O\left(\sum_{u \in D} c_u + mN \log N + m^2 K_{\mathrm{top}}\right). \tag{19}$$

Here $K_{\mathrm{top}} = 10$, and pairwise Top-K_{top} overlap accounts for the last term. Ridge prediction and residual lookup are small relative to expert inference and full sorting. The implementation retains a full score vector for each invoked expert, using $O(mN)$ memory; streaming fusion could reduce this to $O(N)$.