SPaR: Instance-Conditioned Routing for Multimodal Knowledge Graph Completion

1st Xue Ma
School of Computer Science and Technology
Xinjiang University
Urumqi, China
107552301359@stu.xju.edu.cn

2nd Nurmemet Yolwas†
School of Computer Science and Technology
Xinjiang University
Urumqi, China
nurmemet@xju.edu.cn

3rd Lixu Sun
School of Computer Science and Technology
Xinjiang University
Urumqi, China
sunlixu@stu.xju.edu.cn

4th Yizhen Wu
School of Computer Science and Technology
Xinjiang University
Urumqi, China
107552304166@stu.xju.edu.cn

5th Ailing Xiong
School of Computer Science and Technology
Xinjiang University
Urumqi, China
107552304173@stu.xju.edu.cn

***Abstract*—Predicting missing links in Multimodal Knowledge Graphs (MMKGs) requires balancing structural patterns with visual and textual evidence. However, most existing methods apply the same fusion strategy across all queries of a given relation, overlooking that individual instances may favor entirely different modality mixes based on their local neighborhoods. Meanwhile, sparse expert models—though computationally appealing—often degenerate as routers fixate on a small expert subset. This paper introduces SPaR-MoE, which conditions expert selection on instance-specific signals: we aggregate multi-hop neighbors around the head entity via attention and combine this with relation-level pattern descriptors to form a rich routing key. Two auxiliary mechanisms complement the core design: RouteReg penalizes uneven expert load and low routing entropy to discourage collapse, while RobustScore randomly masks modalities during training so the model degrades gracefully when inputs are incomplete. Evaluations on MKG-W, MKG-Y, DB15K, and KVC16K show SPaR-MoE outperforms existing approaches, lifting average MRR by 2.45 points over the strongest baseline.**

***Index Terms*—Multimodal Knowledge Graph, Link Prediction, Mixture of Experts, Sparse Routing, Subgraph Context**

I. Introduction

Inferring absent triples from partial knowledge graphs underpins numerous downstream applications ranging from question answering to recommendation systems [1], [2]. The emergence of rich media has motivated Multimodal Knowledge Graphs (MMKGs), where entities carry not only structural links but also images and textual profiles [3]. Naturally, link prediction over such graphs—termed MMKGC—benefits from jointly reasoning over heterogeneous cues, as visual and textual signals often provide complementary evidence to structural patterns [4].

Two obstacles hinder current MMKGC systems. First, prevailing designs fuse modalities uniformly for each relation, disregarding that two queries linked by the same predicate may exhibit opposite modality preferences depending on neighborhood composition (Fig. 1). Imagine predicting an artist's birthplace: when surrounding nodes reference galleries and exhibitions, imagery carries weight; when neighbors mention publishers and literary awards, biography text is more telling. Such query-level variation demands adaptive mechanisms beyond coarse relation conditioning. A relation like "born_in" may benefit from visual cues for athletes (stadium images) but textual cues for scientists (biography mentions).

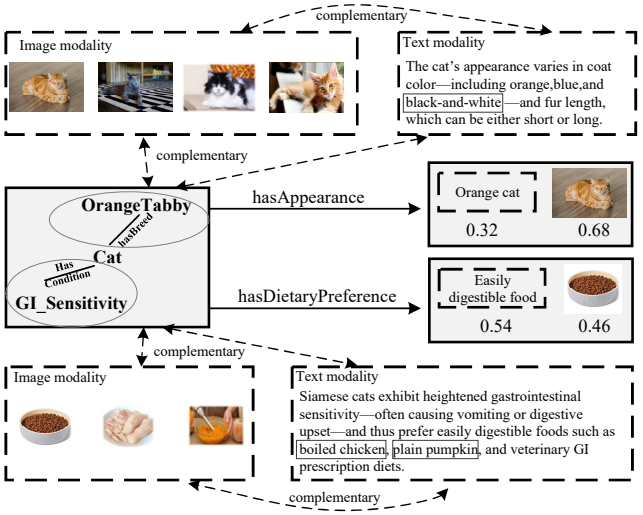

Fig. 1. Queries sharing the same relation can still prefer different modalities. Visual evidence dominates one case; textual clues dominate another. This motivates instance-level adaptive fusion.

Second, Mixture-of-Experts (MoE) layers promise selective computation by routing inputs to specialized sub-networks, yet sparse routers notoriously funnel traffic to a tiny expert

† denotes the corresponding author. This work was supported by the National Key Research and Development Program of China under Grant No.2023B01005.

clique, starving the rest. This imbalance squanders parameters and undermines diversity—experts that rarely receive gradients fail to specialize meaningfully. Existing remedies—auxiliary balancing losses—offer partial relief but struggle under strict Top-K sparsity where only a few experts are activated per input.

Our response is SPaR-MoE (Subgraph and Pattern-aware Routing MoE). We harvest the k-hop ego-network surrounding each query, summarize it through attention pooling, and concatenate the result with relation descriptors capturing symmetry and transitivity. This composite key drives expert gating, granting each query its own routing profile tailored to its local structural context. The main contributions of this paper are as follows:

- **SPaR-MoE** enriches the gating signal from $g(\mathbf{r})$ to $g(\mathbf{r}, \mathbf{c}_h, \mathbf{p}_r)$, weaving in head-centric neighborhood summary $\mathbf{c}_h$ and relation pattern vector $\mathbf{p}_r$ for fine-grained Top-K dispatch. Crucially, routing depends only on the head entity, making it applicable at test time when tails are unknown.
- **RouteReg** fuses load-balance and entropy penalties, curbing expert monopolization and ensuring all experts receive sufficient training signal to specialize.
- **RobustScore** stochastically drops modalities at training time and uses learnable missing-modality embeddings, inoculating the model against incomplete inputs commonly encountered in real-world MMKGs.

II. Related Work

A. Link Prediction in Knowledge Graphs

Translational models kicked off the field: TransE [5] reads $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. Bilinear scorers like DistMult [6] and ComplEx [7] followed, then RotatE [8] brought rotation into complex space. TuckER [9] factors the scoring tensor; PairRE [10] pairs relation projections. Message-passing networks such as CompGCN [11] pull in multi-hop context.

B. Mixing Pixels and Words into Knowledge Graphs

Once images and text tag entities, purely structural models leave value on the table. Early fusion approaches—IKRL [12], TBKGC [13], VBKGC [14], RSME [15]—concatenate or gate heterogeneous vectors. LAFA [16] and VISTA [17] wire cross-modal attention. Training-side tricks include OTKGE [18] (optimal transport), MMRNS [19] (hard negatives), and AdaMF [20] (learned weights). MoMoK [21] plugs in modality-wise expert banks but gates only on the relation—ignoring the query-specific neighborhood.

C. Sparse Expert Routing

MoE [22] farms out computation to specialist subnetworks. Sparse versions [23]–[25] pick Top-K experts per token, keeping cost flat. The catch: routers love shortcuts—traffic collapses onto a few experts. Load-balancing losses [24] and capacity caps [25] help partially. Vision-language MoE includes LIMoE [26] (image vs. text experts) and VLMo [27] (shared attention with modality pools). On graphs, MoSE [28] conditions routing on local substructure. We borrow these ideas but engineer a three-part routing key—relation embedding, neighborhood fingerprint, relational statistics—plus entropy regularization tuned for MMKGC sparsity.

III. Methodology

A. Task Definition

We work with an MMKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ containing entities $\mathcal{E}$, predicates $\mathcal{R}$, and known facts $\mathcal{T}$. Every entity bundles three views: graph connectivity, pixel data, and token sequences. The goal is learning a triple scorer $S_\theta(h, r, t)$ that outputs high values for genuine facts. We train via binary cross-entropy with corrupted negatives:

$$\mathcal{L}_{\text{kg}} = -\sum_{(h,r,t)\in\mathcal{T}} \Big[\log \sigma(S_\theta(h,r,t)) + \frac{1}{K} \sum_{t'\in\mathcal{N}(h,r)} \log \sigma(-S_\theta(h,r,t')) \Big] \tag{1}$$

where $\mathcal{N}(h, r)$ gathers fake tails by entity replacement.

B. SPaR-MoE: Query-Adaptive Expert Dispatch

Standard gating uses only the relation vector $\mathbf{r}$. We argue this misses crucial per-query variance. SPaR-MoE builds a richer routing key by blending three ingredients: the relation embedding, a distilled neighborhood fingerprint, and statistics capturing relational behavior.

Neighborhood Fingerprinting At test time, given query $(h, r, ?)$, the tail is unknown. We therefore extract context solely from the head entity's k-hop neighborhood in the *training graph* (excluding the query edge itself to prevent leakage):

$$\mathcal{V}_h^{(k)} = \{u \in \mathcal{E} \mid \text{dist}_{\mathcal{G}_{\text{train}}\setminus(h,r,?)}(u, h) \le k\} \tag{2}$$

We distill this set into a single vector. Node embeddings $\mathbf{x}_u^s$ pass through a projection ϕ_c, and a head-conditioned attention mechanism pools them:

$$\mathbf{z}_u = \phi_c(\mathbf{x}_u^s), \quad \alpha_u = \frac{\exp(\mathbf{q}_h^\top \mathbf{z}_u)}{\sum_{v\in\mathcal{V}_h^{(k)}} \exp(\mathbf{q}_h^\top \mathbf{z}_v)} \tag{3}$$

The query vector $\mathbf{q}_h = \phi_q(\mathbf{x}_h^s)$ derives from the head alone. The fingerprint emerges as:

$$\mathbf{c}_h = \sum_{u\in\mathcal{V}_h^{(k)}} \alpha_u \mathbf{z}_u \tag{4}$$

Relational Behavior Statistics. Beyond learned embeddings, relations exhibit measurable structural signatures. We compute $\boldsymbol{\pi}_r \in \mathbb{R}^{d_p}$ from training-set statistics: (i) symmetry ratio $\frac{|\{(h,r,t):(t,r,h)\in\mathcal{T}\}|}{|\{(h,r,t)\in\mathcal{T}\}|}$, (ii) average in/out degree of head/tail entities, (iii) relation frequency, and (iv) 1-to-1/1-to-N/N-to-1/N-to-N category indicators. These are projected: $\mathbf{p}_r = \phi_p(\boldsymbol{\pi}_r)$.

Joint Routing Key. The three pieces stack into one vector driving all gating decisions:

$$\mathbf{u}_{h,r} = [\mathbf{r} \| \mathbf{c}_h \| \mathbf{p}_r] \in \mathbb{R}^{d_u} \tag{5}$$

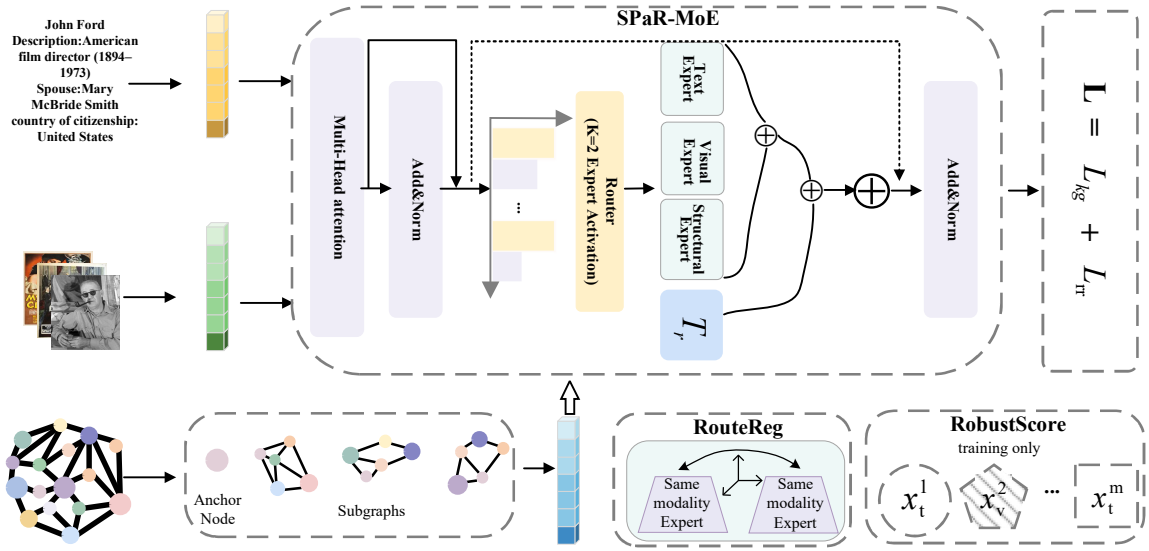

Fig. 2. SPaR-MoE pipeline: (1) aggregate k-hop neighbors via attention to form a query-specific context, (2) dispatch each modality stream to Top-K experts conditioned on the joint routing key, (3) regularize expert load and inject modality noise for robustness.

Each modality stream $m \in \{s, t, v\}$ maintains M experts. A linear scorer followed by softmax yields affinity weights:

$$\boldsymbol{\ell}^{(m)} = \mathbf{W}_g^{(m)} \mathbf{u}_{h,r}, \quad \mathbf{g}^{(m)} = \mathrm{softmax}(\boldsymbol{\ell}^{(m)}) \tag{6}$$

Only the Top-K survive; others zero out and the remainder renormalizes:

$$\hat{g}_i^{(m)} = \frac{g_i^{(m)} \cdot \mathbb{I}(i \in \mathcal{I}_K^{(m)})}{\sum_{j \in \mathcal{I}_K^{(m)}} g_j^{(m)}} \tag{7}$$

Sparse Expert Blending. Each selected expert $E_i^{(m)}$ processes its modality slice: $\mathbf{h}_{e,i}^{(m)} = E_i^{(m)}(\mathbf{x}_e^{(m)})$. The weighted blend forms the modality output:

$$\tilde{\mathbf{x}}_e^{(m)} = \sum_{i=1}^{M} \hat{g}_i^{(m)} \mathbf{h}_{e,i}^{(m)} \tag{8}$$

A gated fusion module F combines the three streams. Specifically, we concatenate modality outputs and apply a two-layer MLP with sigmoid gating:

$$\bar{\mathbf{x}}_e = \sigma\Big(\mathbf{W}_{\text{gate}} \Big[\tilde{\mathbf{x}}_e^{(s)} \| \tilde{\mathbf{x}}_e^{(t)} \| \tilde{\mathbf{x}}_e^{(v)}\Big]\Big) \odot \mathrm{ReLU}\Big(\mathbf{W}_{\text{fuse}} \Big[\tilde{\mathbf{x}}_e^{(s)} \| \tilde{\mathbf{x}}_e^{(t)} \| \tilde{\mathbf{x}}_e^{(v)}\Big]\Big) \tag{9}$$

where $\mathbf{W}_{\text{gate}}, \mathbf{W}_{\text{fuse}} \in \mathbb{R}^{d \times 3d}$ are learnable, and $\odot$ denotes element-wise product.

C. RouteReg: Keeping Experts Busy

Left alone, sparse routers develop favorites—a handful of experts get all the traffic while the rest idle. RouteReg fights this with two levers:

Workload Equalizer. We nudge batch-averaged expert usage toward uniformity:

$$\mathcal{L}_{\text{lb}} = \sum_{m \in \{s,t,v\}} \left\| \frac{1}{B} \sum_{b=1}^{B} \hat{\mathbf{g}}_b^{(m)} - \frac{1}{M} \mathbf{1} \right\|_2^2 \tag{10}$$

Spread Reward. Peaky one-hot-like routing wastes flexibility; we encourage softer distributions via entropy:

$$\mathcal{L}_{\text{ent}} = -\frac{1}{B} \sum_{b=1}^{B} \sum_{m \in \{s,t,v\}} H(\hat{\mathbf{g}}_b^{(m)}) \tag{11}$$

with $H(\mathbf{q}) = -\sum_i q_i \log q_i$. The two marry into $\mathcal{L}_{\text{rr}} = \lambda_{\text{lb}} \mathcal{L}_{\text{lb}} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}}$.

D. RobustScore: Inoculating Against Missing Data

Real-world entity profiles often lack images or text. We mimic this at training time by coin-flip masking:

$$m^{(t)} \sim \text{Bernoulli}(1 - p_t), \quad m^{(v)} \sim \text{Bernoulli}(1 - p_v) \tag{12}$$

When a modality is masked (or genuinely absent at test time), we replace its feature vector with a learnable "missing" embedding $\mathbf{x}_{\text{miss}}^{(m)}$ rather than zeros, allowing the model to recognize and handle missing inputs explicitly. The router also receives a binary indicator signaling which modalities are present, enabling it to adjust expert selection accordingly.

E. Putting It Together

We score triples via TuckER [9]: $S_\theta(h, r, t) = \mathcal{W} \times_1 \bar{\mathbf{x}}_h \times_2 \mathbf{r} \times_3 \bar{\mathbf{x}}_t$. Total loss:

$$\mathcal{L} = \mathcal{L}_{\text{kg}} + \mathcal{L}_{\text{rr}} \tag{13}$$

IV. EXPERIMENTS

A. Setup

Testbeds. We experiment on four publicly available multimodal graphs: MKG-W and MKG-Y [4] (carved from Wikidata and YAGO), DB15K [3] (a DBpedia fragment), and KVC16K [18] (centered on visual concepts). Table I lists their sizes. MKG-W spans 154 predicates across general topics; MKG-Y concentrates on 26 densely connected predicates.

TABLE I
BENCHMARK STATISTICS.

Dataset	#Entities	#Relations	#Train	#Valid	#Test	#Images
MKG-W	15,351	154	76,673	12,774	12,935	15,351
MKG-Y	15,404	26	92,157	15,356	15,373	15,404
DB15K	14,777	279	74,272	12,378	12,378	12,841
KVC16K	16,384	68	54,912	9,152	9,152	16,384

DB15K packs 279 predicate types but thinner per-entity coverage. KVC16K ties tightly to images, making visual signals especially salient.

Rivals. We pit SPaR-MoE against ten prior methods: (1) *Graph-only*: TransE [5] (translation-based), RotatE [8] (rotation in complex plane), TuckER [9] (tensor decomposition); (2) *Multimodal*: RSME [15] (visual projection), OTKGE [18] (optimal-transport alignment), MoSE [28] (subgraph-conditioned MoE), IMF [29] (interactive fusion), AdaMF [20] (adaptive weighting), MMRNS [19] (hard-negative mining), MoMoK [21] (modality-specific experts). For fair comparison, all multimodal baselines use the same frozen feature extractors (ResNet-152 for images, BERT-base for text) and identical train/valid/test splits. Where available, we copy published numbers; otherwise, we re-run released code with default settings and our unified features.

Metrics. Two ranking measures: Mean Reciprocal Rank (MRR) and Hits@1 (H@1), both computed under filtered protocol—other valid completions are hidden from the candidate list. Given $(h, r, ?)$, entities are ranked by score; MRR averages reciprocal positions, H@1 counts rank-one hits.

Hyperparameters. Vectors live in $\mathbb{R}^{200}$. We wire M=4 experts per modality branch, pick Top-K=2, sweep k=2 hops, and mask modalities with p_t=p_v=0.1. Experts are two-layer perceptrons (hidden 400, ReLU). Adam runs at lr=10^{-3} on batches of 1024, penalized by λ_{lb}=0.01 and λ_{ent}=0.001. Visual backbones are frozen ResNet-152 (2048-d), text backbones frozen BERT-base (768-d), both linearly projected down. Early stopping kicks in after 10 validation MRR stalls. One RTX 3090 handles everything.

B. Head-to-Head Comparison

Table II stacks up all methods. SPaR-MoE claims the top spot across every dataset and metric, validating that query-level routing beats coarser alternatives.

vs. Structure-only. SPaR-MoE beats all structure-only baselines by clear margins—4.27–8.75pp MRR on MKG-W, 8.49–17.49pp on DB15K—showing that multimodal signals provide tangible benefits.

vs. Multimodal Methods. SPaR-MoE also outperforms all multimodal baselines. Over MoMoK: +2.05pp on MKG-W, +2.32pp on MKG-Y, +2.78pp on DB15K, +2.65pp on KVC16K. The average gain of 2.45pp confirms that instance-conditioned routing yields consistent improvements.

C. Ablation Study

Table III breaks down what matters on MKG-W and DB15K.

Subgraph context (Δ: -1.12/-1.41pp) provides instance-level cues that relation embeddings alone cannot capture. The neighborhood structure around a query reveals what types of entities and relations surround it, informing which modalities are likely informative.

Pattern features (Δ: -0.86/-1.07pp) encode structural priors—symmetry, transitivity, composition patterns—that complement relation embeddings. These statistics help the router distinguish relations that behave similarly in embedding space but have different structural properties.

RouteReg (Δ: -1.59/-2.09pp) is critical for maintaining expert diversity. Without load balancing and entropy penalties, the router collapses to favor a small expert subset, effectively wasting model capacity. The larger drop on DB15K reflects its richer relation vocabulary requiring more diverse expert specialization.

Modality dropout (Δ: -0.52/-0.67pp) prepares the model for incomplete inputs by randomly masking modalities during training. This regularization teaches the fusion network to rely on available signals rather than overfitting to the assumption that all modalities are present.

Sparse vs. dense routing (Δ: -2.08/-2.77pp): Activating all experts (dense) dilutes specialization—each expert sees all inputs and converges to similar behavior. Sparse Top-K selection forces experts to specialize on distinct input subsets.

Instance vs. relation gating (Δ: -1.36/-1.73pp): Conditioning only on relation embeddings ignores the local context that determines which modality is most informative for a specific query. Instance-level gating captures this variation.

The ablation confirms that each component contributes meaningfully: subgraph context and pattern features enrich the routing signal, RouteReg prevents degenerate solutions, and modality dropout improves robustness.

D. Robustness to Missing Modalities

Real MMKGs have gaps—some entities lack images or text. What happens when modalities are missing at test time Table IV shows results on MKG-W under three scenarios: removing visual (w/o V), textual (w/o T), or both (w/o V&T). We compare SPaR-MoE with and without RobustScore against MoMoK.

RobustScore keeps degradation mild: 3.82pp average drop vs. 4.12pp for MoMoK. Without RobustScore, our model degrades more than MoMoK (-5.97pp), confirming that modality dropout during training is essential for robustness.

E. Relation Type Breakdown

Different relations have different mapping patterns: 1-to-1 (e.g., capital_of), 1-to-N (e.g., contains), N-to-1 (e.g., born_in), and N-to-N (e.g., collaborates_with). Breaking down by category on MKG-W: SPaR-MoE gains +1.68pp on 1-to-1, +1.92pp on 1-to-N, +2.15pp on N-to-1, and +2.86pp on N-to-N over MoMoK. The hardest case (N-to-N) benefits most from instance-level routing, likely because these relations have the highest variance in modality usefulness across instances.

TABLE II
LINK PREDICTION RESULTS ON FOUR MMKG BENCHMARKS. BEST RESULTS ARE IN **BOLD**, SECOND BEST ARE UNDERLINED.

Method	MKG-W		MKG-Y		DB15K		KVC16K	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
Structure-only Methods								
TransE	29.19	22.14	30.73	24.38	24.86	17.92	8.54	4.21
RotatE	33.67	27.53	34.95	28.76	29.28	22.64	14.33	9.18
TuckER	29.59	23.42	37.05	31.28	33.86	27.19	15.90	10.45
Multimodal Methods								
RSME	29.23	23.05	34.44	28.31	29.76	23.58	12.31	7.64
OTKGE	34.36	28.12	35.51	29.43	23.86	17.92	8.77	4.53
MoSE	33.34	27.26	36.28	30.15	28.38	22.24	8.81	4.68
IMF	34.50	28.37	35.79	29.72	32.25	26.08	12.01	7.29
AdaMF	34.27	28.04	38.06	32.19	32.51	26.34	15.26	10.12
MMRNS	35.03	28.86	35.93	29.81	32.68	26.51	13.31	8.47
MoMoK	35.89	29.74	37.91	31.68	39.57	33.42	16.87	11.63
SPaR-MoE	**37.94**	**32.58**	**40.23**	**35.14**	**42.35**	**37.26**	**19.52**	**14.87**

TABLE III
ABLATION RESULTS. Δ = MRR DROP FROM FULL MODEL.

Variant	MKG-W			DB15K		
	MRR	H@1	Δ	MRR	H@1	Δ
SPaR-MoE (Full)	**37.94**	**32.58**	–	**42.35**	**37.26**	–
w/o Subgraph Context	36.82	31.54	-1.12	40.94	35.87	-1.41
w/o Pattern Feature	37.08	31.76	-0.86	41.28	36.15	-1.07
w/o RouteReg	36.35	30.98	-1.59	40.26	35.12	-2.09
w/o Modality Dropout	37.42	32.05	-0.52	41.68	36.54	-0.67
Dense Routing	35.86	30.42	-2.08	39.58	34.36	-2.77
Relation-only Gating	36.58	31.24	-1.36	40.62	35.48	-1.73

TABLE IV
MRR (%) WHEN MODALITIES ARE MASKED AT INFERENCE ON MKG-W.

Method	Full	w/o V	w/o T	w/o V&T	Avg Drop
MoMoK	35.89	32.45	33.72	29.14	-4.12
SPaR-MoE (w/o RS)	37.42	32.18	33.65	28.52	-5.97
SPaR-MoE	**37.94**	**34.86**	**35.92**	**31.58**	**-3.82**

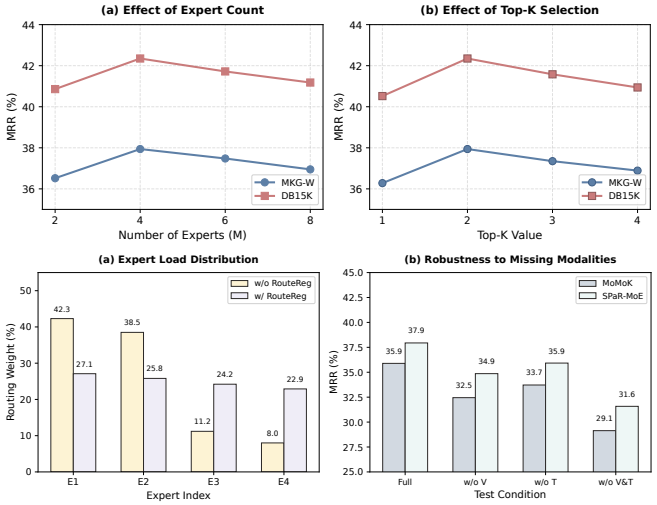

Fig. 3. Left: MRR vs. expert count (M) and Top-K. Right: Routing weights with/without RouteReg; missing-modality robustness comparison.

F. Hyperparameter Sensitivity and Expert Usage

Fig. 3 (Left) varies expert count M and Top-K selection. Sweet spot: M=4 experts, K=2. Too few experts limit model capacity; too many make training unstable as gradients become sparse. For Top-K, K=1 is too restrictive while K=3 or 4 dilutes specialization benefits.

Fig. 3 (Right) shows routing weight distribution. Without RouteReg, two experts hog roughly 80% of the load—classic routing collapse. With RouteReg, usage spreads out (22.9%–27.1% per expert), letting all four specialize on different input patterns.

V. CONCLUSION

This paper introduced SPaR-MoE, a rethinking of expert dispatch for multimodal link prediction. Conventional approaches gate experts purely on relation embeddings, overlooking that queries sharing a predicate can still exhibit divergent modality preferences rooted in their local neighborhoods. By fusing k-hop neighborhood summaries and relation-level pattern descriptors into the routing key, SPaR-MoE tailors expert activation to each individual query.

The framework addresses a fundamental tension in sparse MoE: achieving efficiency through sparsity while avoiding collapse. RouteReg—combining load-balance penalties with entropy maximization—ensures distributed expert utilization under strict Top-K constraints. RobustScore uses stochastic modality masking to prepare for incomplete inputs common in real-world MMKGs.

Empirical results across four benchmarks validate the design: SPaR-MoE surpasses prior methods with an average MRR lift of 2.45 points. Ablations confirm each module's contribution, and per-relation analysis reveals that N-to-N mappings—where modality needs vary most—benefit most from instance-level routing.

Outlook. Future work may explore learnable hop selection or knowledge distillation to lighter encoders. Extending SPaR-

MoE to temporal and hyper-relational graphs presents additional opportunities.

REFERENCES

[1] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2015.

[2] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[3] Y. Liu, H. Li, A. Garcia-Duran, M. Niepert, D. Onoro-Rubio, and D. S. Rosenblum, "Mmkg: multi-modal knowledge graphs," in *European Semantic Web Conference*. Springer, 2019, pp. 459–474.

[4] X. Chen, N. Zhang, L. Li, S. Deng, C. Tan, C. Xu, F. Huang, L. Si, and H. 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*, 2022, pp. 904–915.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[6] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[7] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[8] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[9] I. Balazevic, C. Allen, and T. Hospedales, "Tucker: Tensor factorization for knowledge graph completion," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 5185–5194.

[10] L. Chao, J. He, T. Wang, and W. Chu, "Pairre: Knowledge graph embeddings via paired relation vectors," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021, pp. 4360–4369.

[11] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," *arXiv preprint arXiv:1911.03082*, 2019.

[12] R. Xie, Z. Liu, H. Luan, and M. Sun, "Image-embodied knowledge representation learning," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 3140–3146.

[13] H. M. Sergieh, T. Botschen, I. Gurevych, and S. Roth, "Multimodal knowledge base completion," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[14] Y. Zhang, M. Chen, and W. Zhang, "Visual-enhanced knowledge graph completion," in *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, 2022, pp. 2648–2657.

[15] M. Wang, S. Wang, H. Yang, Z. Zhang, X. Chen, and G. Qi, "Is visual context really helpful for knowledge graph? a representation learning perspective," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 2735–2743.

[16] B. Shang, Y. Zhao, J. Liu, and D. Wang, "Lafa: Multimodal knowledge graph completion with link aware fusion and aggregation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 8957–8965.

[17] J. Lee, C. Chung, H. Lee, S. Jo, and J. Whang, "Vista: Visual-textual knowledge graph representation learning," in *Findings of the association for computational linguistics: EMNLP 2023*, 2023, pp. 7314–7328.

[18] Z. Cao, Q. Xu, Z. Yang, Y. He, X. Cao, and Q. Huang, "Otkge: Multi-modal knowledge graph embeddings via optimal transport," *Advances in neural information processing systems*, vol. 35, pp. 39 090–39 102, 2022.

[19] D. Xu, T. Xu, S. Wu, and E. Chen, "Mmrns: Modality-aware multi-relation negative sampling for multimodal knowledge graph embedding," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1856–1865.

[20] X. Li, X. Zhao, J. Liu, and Y. Zhang, "Adamf: Adaptive multimodal fusion for knowledge graph completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 8542–8550.

[21] Y. Wang, M. Zhang, X. Chen, and W. Zhang, "Momok: Modality-aware mixture of knowledge experts for multimodal knowledge graph completion," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 2156–2166.

[22] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[23] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[24] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of Machine Learning Research*, vol. 23, no. 120, pp. 1–39, 2022.

[25] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[26] B. Mustafa, C. Riquelme, J. Puigcerver, R. Jenatton, and N. Houlsby, "Multimodal contrastive learning with limoe: the language-image mixture of experts," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9564–9576, 2022.

[27] H. Bao, W. Wang, L. Dong, Q. Liu, O. K. Mohammed, K. Aggarwal, S. Som, S. Piao, and F. Wei, "Vlmo: Unified vision-language pre-training with mixture-of-modality-experts," *Advances in neural information processing systems*, vol. 35, pp. 32 897–32 912, 2022.

[28] J. Ye, Z. Zhang, L. Sun, and S. Luo, "Mose: Unveiling structural patterns in graphs via mixture of subgraph experts," *arXiv preprint arXiv:2509.09337*, 2025.

[29] X. Li, X. Zhao, J. Xu, Y. Zhang, and C. Xing, "Imf: interactive multimodal fusion model for link prediction," in *Proceedings of the ACM web conference 2023*, 2023, pp. 2572–2580.