SOHL: SEMI-ORDERED HYPEREDGE REPRESENTATION LEARNING FOR KNOWLEDGE HYPERGRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Traditional hypergraph representation learning for knowledge hypergraphs typically treats hyperedges as unordered collections of entities, failing to capture the critical sequential dependencies and directional semantics embedded within them. Conversely, existing ordered methods rely on rigid global entity orderings, rendering them brittle to noise and positional randomness. To strike a balance without the heavy overhead of strict positional encoding, we introduce SOHL (Semi-Ordered Hyperedge Learning), a novel representation learning framework centered on semi-ordered hyperedges. Rather than enforcing a rigid global order, SOHL partitions exchangeable entities into latent logical roles and retains only their valid sequential relations via a Counterfactual-guided Logical Role Mining Mechanism. Theoretical analysis further demonstrates both the strong expressive power and computational efficiency of our semi-ordered formulation. Extensive experiments across 16 benchmark datasets show that SOHL sets new state-of-the-art performance—outperforming top baselines by up to 3.2% in MRR—while demonstrating exceptional robustness against positional perturbations. Code and datasets are available at `https://anonymous.4open.science/r/SSH-1F4D`.

1 INTRODUCTION

Knowledge hypergraphs Fatemi et al. (2021) are knowledge bases designed to store multi-ary (or n-ary) relations involving variable numbers of entities. Compared to traditional knowledge graphs based on Resource Description Framework (RDF) triples Giunti et al. (2021), knowledge hypergraphs capture richer and more complex latent semantics Yang et al. (2026). To unlock these latent semantics, recent work Yadati (2020); Huang et al. (2024; 2026) has turned to hyperedge-centric representation learning Fatemi et al. (2019); Galkin et al. (2020) to capture complex interactions. Early approaches Yadati et al. (2019) simplify n-ary hyperedges into sets of pairwise edges to leverage conventional graph neural networks. However, this graph reduction inevitably causes severe structural information loss and fails to capture higher-order interactions Feng et al. (2019); Carletti et al. (2020). To overcome this limitation, mainstream hypergraph formulations treat each multi-way relation as an integral structural unit Gao et al. (2022), preserving the original arity and joint membership of facts without projection distortion. Nevertheless, to simplify mathematical modeling, most existing integral hypergraph methods treat nodes within a hyperedge as unordered or permutation-equivalent Liu et al. (2021). This design overlooks the fact that entities often play distinct semantic or logical roles Liu et al. (2021); Huang et al. (2026), thereby losing critical semantic information encoded in entity ordering.

Recent ordered approaches Fatemi et al. (2019); Abboud et al. (2020); Zhu et al. (2021; 2023); Yadati (2020); Huang et al. (2024); Galkin et al. (2024); Cui et al. (2024); Huang et al. (2026) argue that encoding the positions of entities is essential for hyperedge representation learning. However, by enforcing a strict global entity sequence Cui et al. (2024), these methods inherently penalize minor entity permutations caused by noise. This strictness is particularly problematic in real-world hypergraphs, where entities are frequently extracted or ingested in arbitrary orders, leading to random position shifts Luo et al. (2024). Consequently, these models suffer from vulnerability to arbitrary ingestion orderings, prohibitive computational overhead Huang et al. (2026), and a tendency to overfit, ultimately resulting in fragile performance.

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.

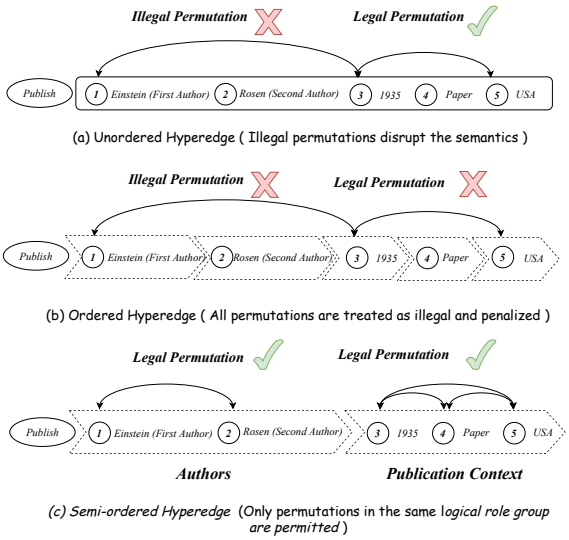


Figure 1: Comparison of hyperedge modeling paradigms on the 5-arity relation Publish. (a) Unordered Hyperedges ignore entity positions, permitting illegal cross-role exchanges (e.g., Einstein $\leftrightarrow$ 1935) that cause severe semantic distortion. (b) Strictly Ordered Hyperedges enforce a rigid global entity sequence, which wrongly penalizes legal intra-role swaps between symmetric entities (e.g., co-authors Einstein $\leftrightarrow$ Rosen) while incurring high computational overhead. (c) Semi-Ordered Hyperedge (SOHL, Ours) achieves both high semantic fidelity and computational efficiency by restricting permutations to legal intra-role exchanges while preventing invalid inter-role permutations.

To address these limitations, we introduce logical roles—relation-specific equivalence classes of entity positions that share identical semantic functions Liu et al. (2021). We observe that permuting entities within the same logical role preserves semantic validity, whereas crossing roles violates relational constraints. This aligns with recent findings Li et al. (2025) showing that entities with equivalent role semantics can be exchanged without altering a knowledge tuple's correctness. For clarity, we define permutations within a logical role as legal and those across different roles as illegal. As illustrated in Figure 1(a), swapping the entities "1935" and "USA" leaves the global semantics of the hyperedge intact; conversely, swapping "Einstein" and "1935" erroneously frames "1935" as an author and "Einstein" as the publication context. Grounded in this insight, we formalize semi-ordered hyperedges, which strictly preserve order only between logical roles rather than across all entities. Based on this formulation, we present SOHL, a novel hyperedge representation learning framework that natively achieves role-level order awareness while remaining invariant to intra-role permutation noise. To accurately assess semantic shifts and reveal these logical roles, we design a Counterfactual-guided Logical Role Mining (CLR) mechanism. Specifically, CLR partitions entity positions into latent, functionally equivalent roles by evaluating their predictive sensitivity under counterfactual position swaps. Theoretically, we prove that the discovered roles and their learned sequence effectively resist cross-role permutations while maintaining high computational efficiency. Extensive experiments across multiple inductive link prediction benchmarks for knowledge hypergraphs demonstrate that SOHL consistently outperforms state-of-the-art baselines, achieving average MRR improvements of 0.1%–3.2% over both ordered and unordered methods. Further empirical analyses, including case studies, confirm that our framework is more robust to illegal permutations while maintaining high computational efficiency.

The key contributions of this paper are summarized as follows:

- We propose SOHL, the first framework to handle semi-ordered hyperedges. By leveraging the logical roles, SOHL addresses the robustness and efficiency issues of existing hypergraph-based inductive link prediction approaches for knowledge hypergraphs.
- We introduce Counterfactual-guided Logical Role Mining to learn latent logical role patterns, supported by theoretical analysis of its robustness and efficiency for multi-arity hyperedges.
- We demonstrate that our SOHL outperforms existing state-of-the-art baselines. Additionally, our case study shows that SOHL offers competitive robustness to illegal permutations.

2 PRELIMINARY

In this section, we provide formal definitions for semi-ordered hypergraphs and inductive link prediction following the framework established by Huang et al. (2026).

Semi-ordered Hypergraph. In the real world, a knowledge hyper-graph G can be denoted as:

$$G = (\mathcal{V}, \mathcal{R}, \mathcal{E}) \tag{1}$$

where $\mathcal{V}$ and $\mathcal{R}$ denote the entity set and relation set, respectively. $\mathcal{E}$ is the set of hyper-edge e, which is formulated as a group of entities sharing with the same relation:

$$e = r(v_1, v_2, \ldots, v_k), \tag{2}$$

where $r \in \mathcal{R}$ is a relation with arity k, and $v_i \in \mathcal{V}$ are entities interacted with r.

Hypergraph-based Inductive Link Prediction. Given an incomplete hyper-edge, the link prediction task aims to infer the missing entity v_i and answer the question q:

$$q = r\ (v_1, \ldots, v_{i-1}, \ ?\ , v_{i+1}, \ldots, v_k), \tag{3}$$

In mainstream literature, the inductive link prediction task Huang et al. (2026) serves as a pretext to train hyperedge representations for knowledge hypergraphs. Under the fully inductive setting, the model is trained on hypergraph $G_{\text{train}} = (\mathcal{V}_{\text{train}}, \mathcal{R}_{\text{train}}, \mathcal{E}_{\text{train}})$ and evaluated on an unseen hypergraph $G_{\text{eval}} = (\mathcal{V}_{\text{eval}}, \mathcal{R}_{\text{eval}}, \mathcal{E}_{\text{eval}})$. Specially, both entities and relations during evaluation are unseen in training process:

$$\mathcal{V}_{\text{train}} \cap \mathcal{V}_{\text{eval}} = \emptyset, \qquad \mathcal{R}_{\text{train}} \cap \mathcal{R}_{\text{eval}} = \emptyset. \tag{4}$$

3 Related Work

Unordered Methods. Early n-ary link prediction methods mainly model hyperedges as unordered collections. NaLP Guan et al. (2019) encodes a hyperedge as a permutation-invariant set of role-value pairs, offering computational efficiency but inevitably admitting illegal cross-role swaps. To preserve role information, RAM Liu et al. (2021) represents each entity together with a latent role, while PosKHG Chen et al. (2023) further incorporates absolute positional features. However, both RAM and PosKHG primarily target transductive completion with a fixed relation vocabulary. Existing surveys similarly categorize n-ary representation learning methods into role-agnostic, position-aware, and role-aware families Lu et al. (2026).

Ordered Methods. Another line of work explicitly treats the argument sequence as part of the fact. Representative approaches include HypE Fatemi et al. (2019), BoxE Abboud et al. (2020), G-MPNN Yadati (2020), HCNet Huang et al. (2024), and ULTRA Galkin et al. (2024), with ULTRA modeling n-ary structures through reification. More recently, HYPER Huang et al. (2026) provides an inductive backbone based on a global entity order. It adopts the 16 JF/WP/MFB/WD splits used in our experiments and demonstrates that perturbing this global order substantially degrades MRR. While such ordered modeling distinguishes cross-role permutations, enforcing a strict global sequence also makes the model sensitive to irrelevant permutations within the same role.

Semi-ordered Methods. Recent studies have begun to relax the two extremes of fully unordered and fully ordered modeling by recognizing that some argument positions may share equivalent semantics. HySAE Li et al. (2025), for example, observes that entities with equivalent role semantics can be exchanged without changing the semantic correctness of a knowledge tuple, and incorporates position-aware role information into hypergraph representations. However, such methods mainly use role or position semantics as auxiliary representations and do not explicitly characterize which permutations should preserve or alter the meaning of a hyperedge.

4 Proposed Model

We propose **SOHL**, a semi-ordered hypergraph representation learning for knowledge hypergraphs as illustrated in Figure 2. It consists of two key components: Counterfactual-guided Logical Role Mining, which discovers logical roles in an unsupervised manner, and Role-aware Hyperedge Encoding, which leverages these discovered roles to encode hyperedges. Below, we detail each component of our method, beginning with the formal notation of logical roles.

4.1 Semi-ordered Hyperedges

To address the sensitivity of prior methods to position permutation, we formally define a semi-ordered hyperedge e as a sequence of logical roles:

$$e = r\{L_1, L_2, \ldots, L_m\},\ L_i \subseteq \mathcal{V}, \tag{5}$$

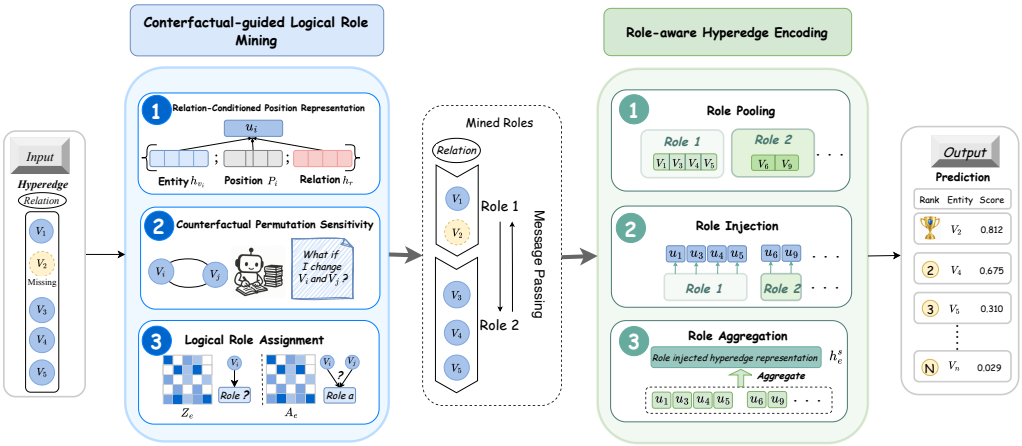


Figure 2: Overall architecture of the SOHL framework, consisting of: **(1) Counterfactual-guided Logical Role Mining**, which learns relation-conditioned position representations and assigns roles via semantic sensitivity under counterfactual permutations (preserving legal intra-role permutations while detecting illegal ones); and **(2) Role-aware Hyperedge Encoding**, which leverages mined roles for role-injected representation and scoring.

where $L_i \subseteq e$ denotes the i-th logical role, subject to $L_i \cap L_j = \emptyset$ for all $i \neq j$, and $\bigcup_{i=1}^{m} L_i = e$.

Expressivity bounds of semi-ordered hyperedges. We establish that this formulation matches the intermediate expressivity bound of ordered and unordered methods (Appendix A).

Based on the concept of logical roles (Eq. 5) and motivated by the observation that entities with equivalent role semantics can be positionally exchanged without altering semantic correctness Li et al. (2025), we define legal and illegal permutations via permutation operators as follows. These definitions serve as the foundation for constructing counterfactuals in Section 4.2.

Legal Permutation. Each logical role L_i contains entities that play identical semantic functions in the hyperedge. Let σ_i denote an intra-role permutation that acts exclusively on entities within L_i while preserving the positions of all other entities. Such a permutation is defined as a *legal permutation*, which satisfies:

$$\sigma_i(e) \equiv e, \qquad \forall, \sigma_i \text{ restricted to } L_i. \tag{6}$$

Illegal Permutation. Conversely, let σ_{cross} denote an inter-role permutation that exchanges entities across distinct logical roles L_i and L_j $(i \neq j)$. Moving entities across different role semantics violates relational constraints and induces an *illegal permutation*, which alters the overall semantics of the hyperedge:

$$\sigma_{\text{cross}}(e) \not\equiv e, \qquad \text{where } \sigma_{\text{cross}} \text{ exchanges entities between } L_i \text{ and } L_j; (i \neq j). \tag{7}$$

Discussion on Semantic Preservation. While Eq. 6 establishes that intra-role permutations theoretically preserve semantic validity, realizing this property in practice requires the model to explicitly capture such invariance. Under our logical-role formulation, because any legal permutation σ_i leaves cross-role assignments intact, the global semantics naturally remains invariant. To enforce or evaluate this property, we leverage counterfactual permutations constructed via σ_i and σ_{cross}. Specifically, following Miyato et al. (2018), we measure the semantic change induced by a legal or illegal permutation as the discrepancy between the predictive distributions of the original hyperedge e and its counterfactual counterpart $\sigma(e)$. Note that alternative metrics, such as representation distances Kornblith et al. (2019), can also be seamlessly integrated.

4.2 Counterfactual-guided Logical Role Mining

Why Counterfactual Permutations? Discovering logical roles is non-trivial because real-world hypergraphs lack explicit role annotations for entities within hyperedges. To learn role equivalence without supervision, a key insight is to evaluate how positional perturbations impact semantic

validity. Counterfactual permutations provide a principled causal intervention mechanism Pearl (2009); Sauer & Geiger (2021): if swapping two entity positions leaves the hyperedge semantics invariant, they are counterfactually identified as sharing the same logical role (legal); otherwise, the exchange induces a semantic discrepancy (illegal). Thus, counterfactual optimization serves as a self-supervised signal You et al. (2020); Miyato et al. (2018) to mine fine-grained logical roles and enforce semantic consistency simultaneously.

Building upon the logical role structure defined in Section 4.1, we formulate this discovery process as a counterfactual-guided optimization over entity positions. Specifically, our objective is to partition entity positions into latent logical roles such that intra-role permutations preserve semantic invariance, while inter-role exchanges exhibit high sensitivity to illegal permutations.

Hyperedge Position Representation. To mine logical roles under a specific relation, we first construct a relation-conditioned representation for each entity position. Given a hyperedge $e = r(v_1, \ldots, v_k)$, the representation of its i-th position u_i is defined as:

$$u_i = [h_{v_i}; p_i; h_r], \tag{8}$$

where h_{v_i} and h_r are the representation of entity v_i and relation r, respectively. p_i is the learnable embedding of the i-th position.

The hyperedge position representation u_i contains the information of the current relation, position, and entity, serving as the evidence for which logical role the entity v_i belongs to. Then we compute the context information of the whole hyperedge h_e based on Equation 8:

$$\mathbf{h}_e = \mathbf{Encoder}\left(\{\mathbf{u}_i\}_{i=1}^{k}\right). \tag{9}$$

where $\mathbf{Encoder}(\cdot)$ can be instantiated by various hyperedge encoders, such as a pooling network Zaheer et al. (2017) used in our implementation or an attention-based encoder.

Counterfactual Permutation Sensitivity (CPS) Following prior counterfactual attribution work that uses KL divergence to quantify intervention-induced changes in predictive distributions Tonekaboni et al. (2020), we define a Counterfactual Permutation Sensitivity (CPS) score for position swaps.

To determine whether two positions i and j are exchangeable, we construct a counterfactual hyperedge $h_{\sigma_{(e)}}$ by swapping their positions. Then both the original and counterfactual hyperedges are fed into the same current model to measure their prediction discrepancy. If i and j share the same logical role, this permutation should leave both the hyperedge semantics and prediction performance largely unaffected. Grounded in this intuition, we formalize the CPS score $\Delta_e(i, j)$ as:

$$\Delta_e(i, j) = \mathrm{KL}\left(P_\theta(\cdot|h_e), P_\theta(\cdot|h_{\sigma_{(e)}})\right), \tag{10}$$

where $h_{\sigma_{(e)}}$ is the counterfactual hyperedge representation, P_θ is the model prediction distribution under h_e, and KL denotes the KL divergence Jeffreys (1946).

Notably, a small $\Delta_e(i, j)$ indicates that swapping entities at positions i and j has minimal impact on both the hyperedge representation and the predictive distribution, implying that these two positions are highly likely to share the same logical role. We also compare our CPS with traditional clustering methods in Section5.2.

Logical Role Assignment. We set the maximum logical-role capacity to M, where each position serves as a candidate role anchor. A direct inclusion of self-exchange would yield $\Delta_e(i, i) = 0$ and may lead to a trivial identity assignment. For the k-th latent logical role, we therefore mask the diagonal CPS entries:

$$\widetilde{\Delta}_e(i, k) = \begin{cases} \Delta_e(i, k), & i \neq k, \\ +\infty, & i = k. \end{cases} \tag{11}$$

The role-assignment probability is then computed as

$$Z_e[i, k] = \frac{\exp\left(-\widetilde{\Delta}_e(i, k)\right)}{\sum_{j=1}^{M} \exp\left(-\widetilde{\Delta}_e(i, j)\right)}. \tag{12}$$

where $Z_e[i,k]$ denotes the soft probability of assigning position i to the k-th latent logical role. The diagonal masking prevents each position from trivially forming an independent singleton role solely through self-exchange.

The learned Z_e will be subsequently passed to the downstream role-aware hyperedge encoding, which governs the soft grouping of positions into logical roles.

4.3 Role-aware Hyperedge Encoding

In this section, we further utilize the learned logical role assignment to encode each hyperedge.

Role Pooling. Leveraging the learned soft assignment matrix Z_e, we aggregate position representations into latent logical role representations. This ensures that positions assigned to the same logical role are consolidated, while distinct logical roles remain separable. Specifically, for the k-th latent logical role, its representation S_k is computed as:

$$S_k = \frac{\sum_{i=1}^{k} Z_e[i,k][\mathbf{u}_i; \mathbf{h}_e]}{\sum_{i=1}^{k} Z_e[i,k]}, \tag{13}$$

For legal permutations within the same logical role, the pooled role representation remains invariant. Conversely, illegal permutations across different logical roles alter the role assignment pattern, thereby yielding distinct logical role representations.

Role Injection. After obtaining the latent logical role representations, we inject the learned logical role semantics back into the hyperedge representation:

$$\mathbf{u_i^s} = \mathbf{u}_i + \lambda \cdot \sum_{k=1}^{M} Z_e[i,k] S_k, \tag{14}$$

where $\mathbf{u}_i$ is the i-th position representation, S_k is the k-th latent logical role representation, and M is the maximum number of roles. λ is a gating factor. Specifically, each position softly retrieves logical role semantics by aggregating all role representations.

Role Aggregation. The retrieved logical role information is then combined with the local position representation $\mathbf{u}_i$ and the global hyperedge context $\mathbf{h}_e$ to produce the logical role-aware hyperedge representation $\mathbf{h}_e^s$:

$$\mathbf{h}_e^s = \mathbf{h}_e + \beta \cdot \sum_{i=1}^{k} \frac{1}{k} \cdot \mathbf{u_i^s}, \tag{15}$$

where $\mathbf{h}_e^s$ denotes the logical role-aware representation of hyperedge e, $\mathbf{h}_e$ is the original hyperedge representation. β is a gating factor.

The logical role injection operation explicitly embeds the learned semi-ordered structure into the final hyperedge representation. Consequently, positions within the same logical role receive consistent role semantics under valid intra-role permutations, whereas cross-role permutations alter the role semantics and modify the representation. This design enables SOHL to preserve intra-role exchangeability while remaining sensitive to invalid cross-role permutations. Furthermore, its computational efficiency is bounded by polynomial complexity (Appendix B).

4.4 Training and Optimization

Following HYPER Huang et al. (2026), given a query hyperedge q, we incorporate its role-aware representation $\mathbf{h}_q^s$ (Eq. 15) into conditional message passing. The prediction module then jointly encodes $\mathbf{h}_q^s$ and candidate entity $\mathbf{h}_v$ to output a compatibility score:

$$\hat{y}_{q,v} = \mathrm{MLP}\big([\mathbf{h}_q^s; \mathbf{h}_v]\big), \tag{16}$$

where a higher $\hat{y}_{q,v}$ indicates a stronger likelihood that entity v satisfies the query q. Our model is subsequently optimized using the binary cross-entropy (BCE) lossLeCun et al. (2015):

$$\mathcal{L} = -\sum_{(q,v,y)} \left[y \log \hat{y}_{q,v} + (1-y) \log(1-\hat{y}_{q,v}) \right], \tag{17}$$

Minimizing $\mathcal{L}$ encourages the model to assign higher probabilities to true entities while penalizing incorrect candidates. Notably, through this unified objective, both the logical role mining and role-aware encoding modules are jointly optimized end-to-end.

5 Experiments and Results

We aim to address the following research questions (RQs).

- **RQ1 (Overall Performance):** To what extent does the SOHL improve reasoning performance compared to unordered and ordered baselines?
- **RQ2 (Ablation Study):** How do the individual components— Counterfactual-guided Logical Role Mining and Role-aware Hyperedge Encoding contribute to the overall reasoning efficacy?
- **RQ3 (Sensitivity):** How do hyperparameters λ and β affect performance?
- **RQ4 (Efficiency):** What is the computational efficiency and scalability of SOHL?
- **RQ5 (Case studies):** Can learned logical role patterns resist illegal permutations?

5.1 Experiment Setup.

Evaluation. Following Huang et al. (2026), we evaluate our model under three settings: ***Zero-shot***: The pretrained model is directly applied to downstream datasets without parameter updates (unseen target entities/relations). ***End-to-End***: The model is randomly initialized and trained solely on each target dataset. ***Fine-tuned***: The pretrained checkpoint is further optimized on the target training split.

Datasets. Following the setting of HYPER Huang et al. (2026), we first pretrain our model with 3 knowledge graphs (KG) and 2 hypergraphs (HG) which are detailed in Appendix D.4, and then evaluate the proposed model on four knowledge hypergraph families, including JF17K Wen et al. (2016) (JF), WikiPeople Guan et al. (2021) (WP), M-FB15K Fatemi et al. (2019) (MFB), and WD50K Galkin et al. (2020) (WD). For each family, four inductive variants are constructed at unseen-relation ratios of 25%, 50%, 75%, and 100%. More dataset statistics appear in Appendix D. Consistent with Huang et al. (2026), the 3KG+2HG mix yields the strongest transfer among the mixtures we test. The overlap-checking procedure is included in the released code.

Permutation Protocol. Cross-role permutations act as synthetic noise retaining original ground-truth targets. MRR measures recovery from corrupted inputs, while KL divergence Jeffreys (1946) measures semantic distortion detection. Implementation details are in Appendix F.

5.2 Results and Analysis.

RQ1: Comparison on ILP Tasks. Table 1 and Figure 3 report inductive MRR and permutation robustness. SOHL achieves overall superior MRR over the second-best baseline HYPER on all 16 end-to-end and 3KG+2HG zero-shot setups (mean gains of $0.009/0.011$), and wins 15 of 16 fine-tuned splits while remaining close to ULTRA.

RQ2: Ablation Results. This section evaluates the efficacy of our key components: Counterfactual-guided Logical Role Mining (CLR) and Role-aware Hyperedge Encoding (RHE). As shown in Table 2, removing either module leads to a clear performance drop, with MRR decreasing by 1.4%–7.9%. Stripping both modules yields a basic backbone variant, further confirming their complementary necessity. Moreover, Figure 3b shows that while KL divergence rises under illegal permutations, the full model maintains a substantially slower MRR decay, demonstrating that CLR and RHE jointly enhance sensitivity to semantic distortion while preserving robust reasoning.

RQ3: Parameter Sensitivity Analysis. We evaluate the impact of the gating factors λ and β across four datasets. As illustrated in Figure 4, SOHL shows consistent sensitivity trends with λ and β, and generally reaches the highest MRR at $\lambda = 0.5$ and $\beta = 0.3$ across all four datasets.

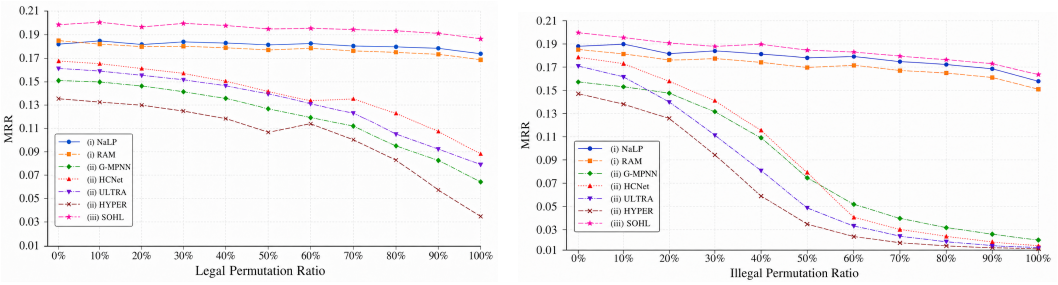
RQ4: Cost Efficiency and Scalability. Figure 5 summarizes our computational efficiency and scalability analysis. As shown in Figure 5a, the training cost of all methods increases with hyperedge

Table 1: MRR results on node- and relation-inductive knowledge hypergraphs (‡ denotes reified models). **Best** and second-best results are highlighted (mean ± std. dev. over 4 runs).

Method	JF				MFB				WP				WD			
	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100
End-to-End Inference																
G-MPNN	.006±.002	.003±.001	.001±.001	.002±.002	.002±.001	.004±.003	.007±.002	.003±.001	.005±.003	.002±.002	.001±.001	.000±.001	.001±.001	.001±.002	.001±.001	.001±.001
HCNet	.011±.003	.009±.002	.069±.006	.028±.004	.033±.005	.026±.003	.016±.004	.082±.007	.104±.009	.050±.006	.019±.004	.003±.002	.086±.008	.043±.005	.015±.003	.007±.002
HYPER	.202±.008	.468±.012	.207±.007	.198±.009	.332±.011	.200±.006	.135±.005	.222±.010	.159±.007	.143±.008	.139±.005	.202±.009	.215±.011	.205±.007	.172±.006	.205±.010
SOHL	**.213±.006**	**.477±.010**	**.223±.009**	**.201±.007**	**.341±.013**	**.207±.008**	**.149±.006**	**.236±.011**	**.164±.007**	**.156±.009**	**.141±.005**	**.209±.008**	**.225±.010**	**.211±.007**	**.188±.009**	**.212±.006**
Zero-shot Inference																
ULTRA‡(3KG)	.103±.006	.437±.013	.168±.008	.144±.007	.255±.011	.235±.009	**.154±.006**	.277±.012	.039±.004	.077±.006	.073±.005	.078±.007	.117±.008	.155±.007	.116±.005	.161±.009
HYPER(3KG)	.148±.007	.297±.011	.112±.006	.130±.008	.248±.010	.191±.007	.039±.004	.276±.013	.143±.006	.147±.008	.186±.009	.221±.010	.167±.007	.158±.006	.123±.008	.146±.005
SOHL(3KG)	.156±.008	.298±.009	.166±.007	.142±.006	.250±.012	.200±.009	.047±.005	.281±.011	**.156±.007**	.153±.006	.171±.008	.210±.010	.170±.009	.152±.007	.120±.006	.155±.008
ULTRA‡(4KG)	.011±.003	.298±.012	.042±.004	.082±.006	.217±.009	.170±.007	.043±.005	.135±.008	.009±.002	.006±.001	.006±.003	.004±.002	.027±.004	.069±.006	.063±.005	.065±.007
HYPER(4KG)	.109±.007	.065±.005	.128±.008	.087±.006	.116±.007	.117±.005	.089±.006	.148±.009	.074±.005	.212±.009	.175±.008	.180±.007	.148±.009	.111±.006	.150±.008	.255±.011
SOHL(4KG)	.111±.006	.078±.005	.131±.009	.090±.006	.122±.008	.139±.007	.092±.005	.142±.009	.095±.006	**.233±.010**	.181±.007	.192±.008	.122±.007	.116±.006	.157±.009	**.270±.012**
ULTRA‡(50KG)	.001±.001	.096±.006	.010±.003	.001±.001	.225±.010	.083±.007	.001±.001	.190±.009	.006±.002	.009±.003	.009±.002	.004±.001	.008±.002	.001±.001	.001±.001	.001±.001
HYPER(50KG)	.056±.005	.294±.011	.084±.006	.111±.007	.122±.008	.156±.007	.126±.009	.148±.006	.067±.004	.198±.010	.191±.008	.155±.007	.073±.005	.055±.004	.130±.008	.088±.006
SOHL(50KG)	.081±.006	.288±.012	.074±.005	.123±.008	.126±.007	.155±.009	.131±.006	.155±.008	.075±.005	.201±.011	.193±.009	.166±.007	.088±.006	.062±.004	.131±.008	.072±.005
ULTRA‡(4HG)	.154±.008	.442±.014	.175±.007	.170±.009	.338±.012	.236±.010	.129±.006	.280±.011	.052±.004	.091±.006	.089±.005	.089±.007	.076±.005	.175±.008	.052±.004	.136±.009
HYPER(4HG)	.187±.009	.377±.013	.188±.008	.181±.007	.349±.012	.244±.009	.139±.006	.278±.011	.075±.005	.068±.004	.086±.006	.168±.008	.087±.006	.158±.007	.057±.005	.165±.009
SOHL(4HG)	.189±.008	.390±.011	.192±.007	**.184±.009**	.351±.013	.260±.010	.137±.006	.278±.012	.094±.006	.071±.005	.098±.007	.170±.009	.084±.006	.167±.008	.060±.004	.174±.007
ULTRA‡(3KG+2HG)	.209±.009	.446±.013	.187±.007	.168±.008	.343±.012	.236±.009	.128±.006	.283±.011	.045±.004	.086±.006	.080±.005	.090±.007	.183±.008	.182±.007	.127±.006	.137±.008
HYPER(3KG+2HG)	.216±.010	.455±.015	.213±.008	.173±.007	.363±.013	.250±.010	.140±.006	.299±.012	.132±.006	.152±.007	.192±.009	.222±.010	.223±.009	.200±.008	.154±.007	.182±.009
SOHL(3KG+2HG)	**.225±.008**	**.470±.012**	**.226±.009**	.177±.007	**.373±.011**	**.259±.009**	.143±.006	**.304±.013**	.137±.006	.166±.008	**.210±.009**	**.245±.011**	**.247±.010**	**.211±.008**	**.160±.007**	.189±.009
Finetuned Inference																
ULTRA‡(3KG+2HG)	.214±.009	.438±.013	.193±.008	.174±.007	**.351±.012**	**.244±.009**	.136±.006	**.291±.011**	.051±.004	.092±.006	.086±.005	.097±.007	.191±.008	.189±.009	.134±.006	.145±.007
HYPER(3KG+2HG)	.217±.010	.456±.014	.209±.008	.176±.007	.347±.012	.243±.010	.158±.007	.275±.011	.169±.007	.171±.008	.194±.009	.210±.010	.225±.009	.234±.010	.166±.008	.210±.009
SOHL(3KG+2HG)	**.221±.008**	**.460±.012**	**.212±.009**	**.183±.007**	.350±.011	.240±.010	**.163±.007**	.281±.012	**.172±.008**	**.174±.007**	**.197±.009**	**.211±.010**	**.227±.009**	**.239±.011**	**.168±.007**	**.219±.010**

Table 2: Ablation results of SOHL with 3KG+2HG.

Method	JF				MFB				WP				WD			
	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100
End-to-End Inference																
SOHL w/o CLR & RHE	.165±.009	.394±.014	.172±.008	.156±.006	.251±.012	.161±.007	.116±.005	.184±.010	.130±.006	.113±.004	.105±.007	.156±.009	.181±.011	.162±.006	.139±.008	.154±.005
SOHL w/o CLR	.178±.007	.412±.013	.186±.009	.170±.006	.305±.011	.185±.008	.128±.005	.200±.009	.147±.006	.125±.004	.117±.007	.171±.008	.196±.010	.176±.006	.162±.009	.170±.005
SOHL w/o RHE	.188±.006	.417±.012	.195±.008	.170±.007	.273±.010	.176±.006	.129±.004	.211±.009	.142±.005	.135±.007	.123±.006	.187±.008	.199±.011	.184±.007	.152±.005	.170±.006
SOHL w/o CPS	.201±.007	.451±.012	.211±.008	.190±.006	.326±.011	.198±.007	.141±.005	.221±.009	.156±.006	.145±.005	.133±.006	.195±.008	.214±.009	.198±.007	.179±.007	.199±.006
SOHL	**.213±.008**	**.477±.014**	**.223±.009**	**.201±.007**	**.341±.012**	**.207±.008**	**.149±.006**	**.236±.010**	**.164±.007**	**.156±.005**	**.141±.006**	**.209±.009**	**.225±.010**	**.211±.007**	**.188±.008**	**.212±.006**
Zero-shot Inference																
SOHL w/o CLR & RHE	.170±.008	.372±.015	.165±.007	.133±.005	.307±.013	.192±.009	.108±.004	.246±.011	.111±.005	.126±.007	.158±.008	.183±.010	.197±.009	.161±.006	.117±.005	.148±.007
SOHL w/o CLR	.188±.008	.415±.012	.182±.006	.150±.007	.333±.011	.231±.009	.122±.005	.264±.010	.123±.006	.143±.007	.186±.008	.198±.009	.214±.010	.178±.006	.138±.005	.162±.007
SOHL w/o RHE	.186±.007	.391±.013	.181±.008	.147±.006	.325±.012	.208±.007	.119±.004	.269±.011	.122±.005	.139±.006	.172±.009	.209±.008	.215±.009	.175±.007	.129±.006	.161±.005
SOHL w/o CPS	.211±.008	.446±.013	.213±.007	.168±.006	.356±.012	.247±.009	.136±.005	.289±.011	.131±.005	.157±.006	.201±.008	.229±.010	.232±.009	.199±.007	.151±.006	.181±.008
SOHL	**.225±.009**	**.470±.015**	**.226±.008**	**.177±.007**	**.373±.013**	**.259±.010**	**.143±.006**	**.304±.012**	**.137±.005**	**.166±.007**	**.210±.009**	**.245±.011**	**.247±.010**	**.211±.008**	**.160±.006**	**.189±.009**
Finetuned Inference																
SOHL w/o CLR & RHE	.161±.007	.383±.014	.166±.006	.144±.005	.276±.012	.186±.008	.132±.006	.218±.010	.136±.005	.131±.006	.149±.007	.160±.009	.168±.008	.179±.010	.133±.005	.172±.007
SOHL w/o CLR	.198±.008	.400±.013	.181±.007	.162±.006	.293±.011	.201±.008	.146±.005	.236±.010	.149±.006	.150±.007	.162±.008	.175±.006	.182±.009	.194±.008	.146±.005	.187±.007
SOHL w/o RHE	.177±.006	.411±.012	.186±.008	.158±.007	.312±.010	.202±.009	.145±.006	.240±.011	.150±.005	.144±.006	.174±.007	.187±.009	.187±.008	.203±.010	.151±.006	.193±.008
SOHL w/o CPS	.210±.008	.437±.012	.201±.007	.174±.006	.332±.011	.226±.008	.156±.006	.268±.010	.163±.006	.166±.006	.187±.008	.198±.009	.214±.008	.225±.009	.160±.006	.207±.008
SOHL	**.221±.009**	**.460±.014**	**.212±.008**	**.183±.007**	**.350±.012**	**.240±.009**	**.163±.006**	**.281±.011**	**.172±.007**	**.174±.006**	**.197±.009**	**.211±.010**	**.227±.009**	**.239±.010**	**.168±.007**	**.219±.009**

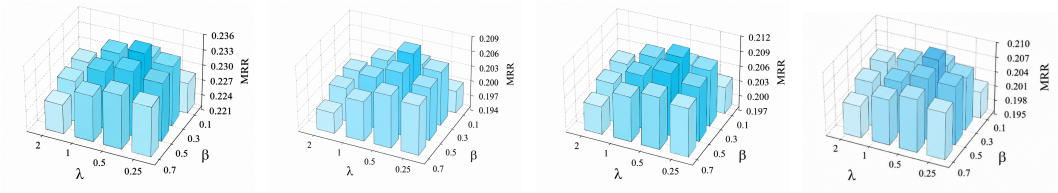


(a) Different performance with legal permutation.

(b) Different performance with illegal permutation.

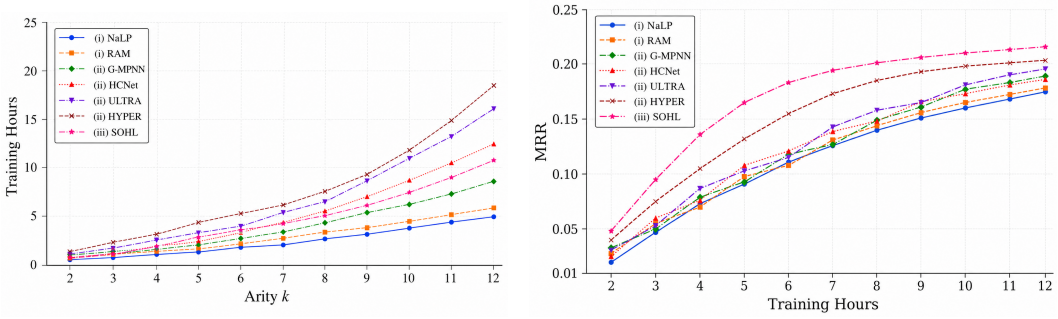
Figure 3: Performance of different baseline methods (**(i) Unordered**, **(ii) Ordered**, **(iii) Semi-ordered**) from 0% to 100% permutation ratio across four datasets of Zero-shot Inference.

arity M, whereas SOHL exhibits a moderate and stable growth trend. Although logical role mining introduces additional computation, SOHL remains more efficient than several ordered baselines, particularly HYPER and ULTRA, whose training overhead grows rapidly for high-arity hyperedges. Figure 5b further shows that SOHL achieves the fastest early-stage convergence and consistently attains the highest MRR under the same training budget.



(a) MFB Dataset. (b) WP Dataset. (c) WD Dataset. (d) JF Dataset.

Figure 4: Sensitivity of gating factor λ and β across four datasets.



(a) Training time with increasing hyperedge arity M. (b) MRR performance with respect to training time.

Figure 5: Average computational efficiency and scalability across four datasets.

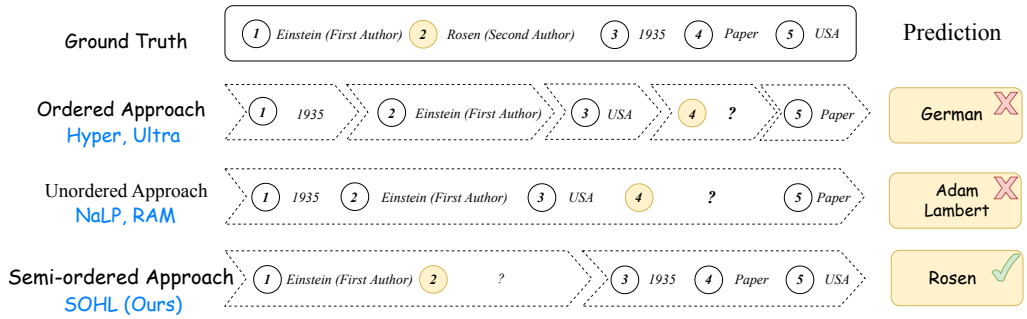


Figure 6: Comparison on the "Publish" hyperedge.

RQ5: Case Study. To evaluate SOHL's ability to mine logical role patterns and resist illegal permutations, we perform a case study on a representative 5-arity hyperedge (Figure 6). Upon an illegal cross-role permutation, the ordered approach predicts incorrectly due to absolute position dependence, while the unordered approach suffers from role ambiguity. In contrast, SOHL correctly separates the *Author* and *Publication Context* roles while preserving within-role exchangeability, successfully predicting *Rosen (Second Author)*. This confirms that SOHL maintains valid within-role invariance while remaining sensitive to cross-role violations.

6 Conclusion

We introduce SOHL, a semi-ordered framework for inductive link prediction on knowledge hypergraphs. SOHL groups exchangeable positions into latent logical roles while retaining positional distinctions across roles, supported by Counterfactual-guided Logical Role Mining and Role-aware Hyperedge Encoding. Theoretically, we establish a polynomial complexity bound (Appendix B) and intermediate expressivity bounds (Appendix A). Extensive experiments demonstrate that SOHL consistently outperforms strong baselines across end-to-end and zero-shot settings, remains highly competitive under fine-tuning, and exhibits superior robustness against illegal position swaps compared to ordered models.

USE OF GENERATIVE AI

We used generative AI tools for language polishing of the manuscript. We have reviewed and verified all AI-assisted text and take full responsibility for the final content of this work, including claims, equations, and experimental descriptions.

LIMITATIONS

This work focuses on inductive reasoning on knowledge hypergraphs, leaving transductive role-aware models (e.g., RAM, PosKHG) outside the current scope. Additionally, our permutation protocol keeps the original target intact to specifically measure robustness against ingestion-order noise. Finally, logical roles are inferred in an unsupervised, data-driven manner via counterfactual swap sensitivity rather than relying on explicit role annotations.

ETHICS STATEMENT

This work studies inductive link prediction on publicly available knowledge hypergraph benchmarks and does not involve human subjects, personally identifiable information, or sensitive applications. We are not aware of specific ethical concerns beyond the standard risks of automated knowledge completion systems.

REPRODUCIBILITY STATEMENT

We release anonymous source code and datasets at `https://anonymous.4open.science/r/SSH-1F4D`.

REFERENCES

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. Boxe: A box embedding model for knowledge base completion. *Advances in neural information processing systems*, 33:9649–9661, 2020.

Timoteo Carletti, Federico Battiston, Giulia Cencetti, and Duccio Fanelli. Random walks on hypergraphs. *Physical review E*, 101(2):022308, 2020.

Zirui Chen, Xin Wang, Chenxu Wang, and Zhao Li. Poskhg: A position-aware knowledge hypergraph model for link prediction: Z. chen et al. *Data Science and Engineering*, 8(2):135–145, 2023.

Yuanning Cui, Zequn Sun, and Wei Hu. A prompt-based knowledge graph foundation model for universal in-context reasoning. *Advances in Neural Information Processing Systems*, 37:7095–7124, 2024.

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.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. Knowledge hypergraphs: Prediction beyond binary relations. *arXiv preprint arXiv:1906.00137*, 2019.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. Knowledge hypergraphs: prediction beyond binary relations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 2191–2197, 2021.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 7346–7359, 2020.

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.

Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.

Marco Giunti, Giuseppe Sergioli, Giuliano Vivanet, and Simone Pinna. Representing n-ary relations in the semantic web. *Logic Journal of the IGPL*, 29(4):697–717, 2021.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. Link prediction on n-ary relational data. In *The world wide web conference*, pp. 583–593, 2019.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. Link prediction on n-ary relational data based on relatedness evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):672–685, 2021.

Xingyue Huang, Miguel Romero Orth, Pablo Barceló, Michael M Bronstein, and Ismail Ilkan Ceylan. Link prediction with relational hypergraphs. *arXiv preprint arXiv:2402.04062*, 2024.

Xingyue Huang, Mikhail Galkin, Michael Bronstein, and Ismail I Ceylan. Hyper: A foundation model for inductive link prediction with knowledge hypergraphs. In *International Conference on Learning Representations*, volume 2026, pp. 126914–126947, 2026.

Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Zhao Li, Xin Wang, Jun Zhao, Feng Feng, Zirui Chen, and Jianxin Li. Hysae: an efficient semantic-enhanced representation learning model for knowledge hypergraph link prediction. In *Proceedings of the ACM on Web Conference 2025*, pp. 86–97, 2025.

Yu Liu, Quanming Yao, and Yong Li. Role-aware modeling for n-ary relational knowledge bases. In *Proceedings of the web conference 2021*, pp. 2660–2671, 2021.

Xiaohua Lu, Liubov Tupikina, and Mehwish Alam. Two-dimensional taxonomy for n-ary knowledge representation learning methods. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Haoran Luo, Yuhao Yang, Tianyu Yao, Yikai Guo, Zichen Tang, Wentai Zhang, Shiyao Peng, Kaiyang Wan, Meina Song, Wei Lin, et al. Text2nkg: Fine-grained n-ary relation extraction for n-ary relational knowledge graph construction. *Advances in Neural Information Processing Systems*, 37:27417–27439, 2024.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Tara Safavi and Danai Koutra. Codex: A comprehensive knowledge graph completion benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8328–8350, 2020.

Axel Sauer and Andreas Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

Sana Tonekaboni, Shalmali Joshi, Kieran Campbell, David K Duvenaud, and Anna Goldenberg. What went wrong and when? instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems*, 33:799–809, 2020.

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.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. *arXiv preprint arXiv:1604.08642*, 2016.

Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 564–573, 2017.

Naganand Yadati. Neural message passing for multi-relational ordered and recursive hypergraphs. *Advances in Neural Information Processing Systems*, 33:3275–3289, 2020.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

Yaming Yang, Ziyu Zheng, Weigang Lu, Zhe Wang, Xinyan Huang, Wei Zhao, and Ziyu Guan. Defining and discovering hyper-meta-paths for heterogeneous hypergraphs. *Advances in Neural Information Processing Systems*, 38:43216–43240, 2026.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

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.

APPENDIX: TABLE OF CONTENT

A Semi-ordered Expressivity **13**

B Computational Complexity **13**

C Scoring Models **14**

D Dataset Details **14**

D.1 Evaluation Dataset Families 14

D.2 Construction of the Inductive Benchmarks 14

D.3 Statistics of the 16 Evaluation Datasets 15

D.4 Pretraining Corpora and Model Variants 15

E Baselines and Pretraining Variants **16**

F Implementation Notes **17**

A SEMI-ORDERED EXPRESSIVITY

Proposition 1 (Semi-ordered Expressivity). Given a valid partition of a k-arity hyperedge into m logical roles with sizes $\{|L_i|\}_{i=1}^m$, the number of distinguishable semantic arrangements is

$$N = \frac{k!}{\prod_{i=1}^m |L_i|!} \in (1, k!). \tag{A.1}$$

The two bounds correspond to fully unordered and fully ordered modeling, respectively.

This result shows that semi-ordered modeling induces an intermediate semantic space between the two extremes. Cross-role exchanges remain distinguishable, whereas permutations among positions assigned to the same logical role are treated as equivalent.

Proof. Fix a hyperedge of arity k together with a partition of its k positions into m logical roles of sizes $\{|L_i|\}_{i=1}^m$, so that $\sum_{i=1}^m |L_i| = k$ and $|L_i| \geq 1$ for every i. Two position assignments represent the same semi-ordered hyperedge if and only if one is obtained from the other by permuting entities inside logical roles. The set of all such intra-role permutations is the product of symmetric groups $\prod_{i=1}^m S_{|L_i|}$, whose cardinality is $\prod_{i=1}^m |L_i|!$.

The symmetric group S_k acts on the k entity positions by permutation. The stabilizer of a given arrangement under intra-role exchange is exactly $\prod_{i=1}^m S_{|L_i|}$. By the orbit–stabilizer theorem, the number of orbits, that is, the number of distinguishable semantic arrangements, is

$$N = \frac{|S_k|}{\prod_{i=1}^m |S_{|L_i|}|} = \frac{k!}{\prod_{i=1}^m |L_i|!}.$$

If every position belongs to a single role, then $m = 1$ and $|L_1| = k$, so $N = 1$, which recovers fully unordered modeling. If every role is a singleton, then $m = k$ and $|L_i| = 1$ for all i, so $N = k!$, which recovers fully ordered modeling. For any proper semi-ordered partition with at least one role of size greater than one and at least two roles, one has $1 < N < k!$. This completes the proof.

B COMPUTATIONAL COMPLEXITY

Proposition 2 (Computational Complexity). Let d denote the representation dimension, $M \leq k$ the maximum logical-role capacity, and C the cost of producing one predictive distribution. The total arithmetic complexity of logical-role mining and role-aware encoding is

$$T = O\left(k^2 C + kMd\right), \qquad S = O\left(kM + Md\right). \tag{B.1}$$

Proof. We account for the two modules of SOHL separately.

Counterfactual-guided logical role mining. For a hyperedge of arity k, Counterfactual Permutation Sensitivity evaluates a position swap (i, j) for each unordered pair $i \neq j$. There are $k(k-1)/2$ such pairs. Each swap constructs a counterfactual hyperedge and computes a KL divergence between two predictive distributions, at cost C per pair. The resulting arithmetic cost is

$$T_{\mathrm{CPS}} = O\left(\frac{k(k-1)}{2}C\right) = O(k^2C). \tag{B.2}$$

The masked CPS matrix and the soft assignment $Z_e \in \mathbb{R}^{k\times M}$ occupy $O(kM)$ storage. Forming Z_e from the $k \times M$ scores costs $O(kM)$ additional arithmetic, which is dominated by T_{CPS} whenever $C = \Omega(1)$.

Role-aware hyperedge encoding. Role pooling computes M role representations, each a d-dimensional weighted average over k positions, at cost $O(kMd)$. Role injection mixes the M role vectors into k position states, again at cost $O(kMd)$. Role aggregation then averages k injected states in $O(kd)$. The encoding cost is therefore

$$T_{\mathrm{enc}} = O(kMd). \tag{B.3}$$

Storing M role vectors uses $O(Md)$ memory, which together with Z_e yields $S = O(kM + Md)$.

Adding the two modules gives $T = T_{\mathrm{CPS}} + T_{\mathrm{enc}} = O(k^2C + kMd)$. The $k(k-1)/2$ counterfactual exchanges can be evaluated in parallel batches, which reduces wall-clock time but does not change the total arithmetic complexity. In particular, SOHL never enumerates the $N = k!/\prod_{i=1}^{m}|L_i|!$ semantic arrangements from Eq. A.1, and the complexity remains polynomial in the hyperedge arity k.

C Scoring Models

NaLP Guan et al. (2019), RAM Liu et al. (2021), HypE Fatemi et al. (2019), BoxE Abboud et al. (2020), and PosKHG Chen et al. (2023) are discussed in Section 3 as unordered, role-aware, or ordered scoring models. They are not re-run on the 16 inductive splits. Table 7 summarizes their arity handling and permutation behavior relative to SOHL.

D Dataset Details

D.1 Evaluation Dataset Families

Following HYPER Huang et al. (2026), we evaluate SOHL on 16 node-and-relation inductive knowledge-hypergraph benchmarks constructed from four source datasets: JF17K (JF), WikiPeople (WP), WD50K (WD), and M-FB15K (MFB). For each source family, the suffix in $\{-25, -50, -75, -100\}$ denotes the target percentage of inference facts whose relations are unseen during training. The 16 benchmarks therefore cover different source domains, graph sizes, relation vocabularies, arities, and degrees of relation shift.

D.2 Construction of the Inductive Benchmarks

We use the benchmark construction protocol released with HYPER Huang et al. (2026). Given a source knowledge hypergraph, the procedure first samples disjoint training and inference relation sets according to the relation-sampling ratio p_{rel}. It then samples training entities and retains the largest connected component induced by their neighboring hyperedges. All selected training entities are removed before constructing the inference graph, ensuring $\mathcal{V}_{\mathrm{train}} \cap \mathcal{V}_{\mathrm{inf}} = \varnothing$. The inference graph is subsequently sampled from the remaining entities and contains seen and unseen relations in the ratio specified by p_{tri}. Its largest connected component is retained and its hyperedges are divided into auxiliary, validation, and test subsets using a 3:1:1 ratio.

Here, n_{train} and n_{inf} are the initial numbers of sampled entities for constructing the training and inference graphs, respectively; p_{rel} controls the relation partition; and p_{tri} controls the proportion of

inference facts instantiated by unseen relations. Because connected components are extracted after sampling, the final numbers of entities and hyperedges need not equal the initial sampling budgets. The generation hyperparameters for the 16 splits are shown in Table 3.

Table 3: Generation hyperparameters for the 16 inductive benchmarks.

Dataset	n_{train}	n_{inf}	p_{rel}	p_{tri}	Dataset	n_{train}	n_{inf}	p_{rel}	p_{tri}
JF-25	1,000	900	0.40	0.25	WD-25	700	1,200	0.25	0.25
JF-50	1,000	800	0.50	0.50	WD-50	1,000	1,000	0.50	0.50
JF-75	1,200	1,200	0.40	0.75	WD-75	10,000	8,000	0.50	0.75
JF-100	1,200	1,200	0.50	1.00	WD-100	10,000	8,000	0.50	1.00
WP-25	900	800	0.40	0.25	MFB-25	100	95	0.50	0.25
WP-50	800	1,000	0.30	0.50	MFB-50	100	85	0.60	0.50
WP-75	1,000	1,000	0.50	0.75	MFB-75	80	80	0.50	0.75
WP-100	1,000	1,000	0.50	1.00	MFB-100	120	100	0.50	1.00

D.3 Statistics of the 16 Evaluation Datasets

The detailed statistical information for the 16 benchmarks is shown in Table 4. The inference graph provides the observed context used to answer test queries, whereas the test split contains the held-out facts used for ranking evaluation. A max-arity entry of the form a/b gives the maximum arity in the training and inference graphs, respectively; a single value indicates that the two maxima are equal.

Table 4: Statistics of node-and-relation inductive knowledge hypergraph benchmarks.

Dataset	Training			Inference			Test			Max Arity
	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	
JF-25	2,616	41	3,371	1,159	36	1,056	209	15	103	5/4
JF-50	2,859	53	3,524	1,102	37	1,292	157	5	109	5
JF-75	3,129	67	4,287	1,488	38	1,697	225	11	131	5
JF-100	2,123	48	2,449	1,696	35	2,159	52	5	25	5
WP-25	6,378	128	7,453	2,784	66	4,794	830	19	959	6/4
WP-50	7,586	155	9,536	3,608	87	4,390	531	29	413	7/6
WP-75	7,787	118	9,271	4,737	101	6,221	629	27	459	6
WP-100	7,787	118	9,271	4,891	63	7,516	275	15	155	6
WD-25	4,533	239	5,482	3,008	191	3,106	250	37	148	22/5
WD-50	3,796	162	4,147	2,303	188	2,353	145	30	91	19/6
WD-75	6,518	243	6,305	5,194	244	5,831	547	57	385	22/5
WD-100	6,798	237	7,271	3,576	105	3,951	385	29	282	19/4
MFB-25	1,266	11	8,182	1,929	12	2,802	146	7	87	3/5
MFB-50	1,415	11	8,409	1,528	13	2,426	472	10	486	3/5
MFB-75	2,225	15	5,271	1,363	16	4,008	675	11	803	3/4
MFB-100	2,013	19	11,658	2,406	5	4,514	808	5	904	3/5

The four source families are as follows:

- JF17K Wen et al. (2016) contains heterogeneous multi-ary facts derived from Freebase.
- WikiPeople Guan et al. (2021) contains person-centered facts extracted from Wikidata and includes both binary and higher-arity relations.
- WD50K Galkin et al. (2020) is a Wikidata-derived hyper-relational benchmark; following HYPER, a main triple and its associated qualifiers are represented jointly as an ordered hyperedge.
- M-FB15K Fatemi et al. (2019) is a multi-ary Freebase benchmark whose relations are predominantly ternary.

D.4 Pretraining Corpora and Model Variants

We follow HYPER's controlled pretraining mixtures to examine transfer across binary knowledge graphs (KGs) and higher-arity knowledge hypergraphs (KHGs). The binary pretraining datasets are FB15k-237 Toutanova & Chen (2015), WN18RR Dettmers et al. (2018), CoDEx-M Safavi & Koutra (2020), and NELL995 Xiong et al. (2017). The hypergraph pretraining datasets are JF17K, WikiPeople, FB-AUTO, and M-FB15K. The 3KG+2HG mixture, used as the principal pretrained checkpoint in our experiments, combines FB15k-237, WN18RR, CoDEx-M, JF17K, and WikiPeople. The pretraining data used by each variant is shown in Table 5.

Table 5: Pretraining data used by different model variants. The same mixtures are used for ULTRA, HYPER, and SOHL unless otherwise stated.

Variant	Pretraining datasets
3KG	FB15k-237, WN18RR, and CoDEx-M
4KG	FB15k-237, WN18RR, CoDEx-M, and NELL995
50KG	FB15k-237, WN18RR, CoDEx-M, NELL995, and the remaining 46 KGs in the ULTRA-50G corpus
4HG	JF17K, FB-AUTO, WikiPeople, and M-FB15K
3KG+2HG	FB15k-237, WN18RR, CoDEx-M, JF17K, and WikiPeople
End-to-End	Only the training graph of the corresponding target benchmark

Table 6 gives the statistics of the named datasets reported by HYPER. For the three binary KGs, every fact has arity two. The arity counts are computed over the union of their training, validation, and test splits.

Table 6: Statistics of the named pretraining knowledge graphs and knowledge hypergraphs.

Statistic	AUTO	WP	JF	MFB	FB	WN	CoDEx
$\lvert\mathcal{V}\rvert$	3,410	47,765	29,177	10,314	14,541	40,943	17,050
$\lvert\mathcal{R}\rvert$	8	707	327	71	237	11	51
Train facts	6,778	305,725	61,104	415,375	272,115	86,835	185,584
Validation facts	2,255	38,223	15,275	39,348	17,535	3,034	10,310
Test facts	2,180	38,281	24,915	38,797	20,466	3,134	10,311
Maximum arity	5	9	6	5	2	2	2
Arity $= 2$	3,786	337,914	56,322	82,247	310,116	93,003	206,205
Arity $= 3$	0	25,820	34,550	400,027	0	0	0
Arity $= 4$	215	15,188	9,509	26	0	0	0
Arity ≥ 5	7,212	3,307	2,267	11,220	0	0	0

Clarification on the 50KG corpus. The HYPER paper reports the additional component of the 50KG mixture only as "Others (46G)", following the ULTRA-50G pretraining corpus, rather than enumerating the 46 dataset names individually. We therefore retain the same designation instead of introducing an unverifiable list. This does not affect the reproducibility of our principal 3KG+2HG setting, whose five constituent datasets are explicitly specified above.

E Baselines and Pretraining Variants

Baselines. To evaluate inductive MRR, we compare SOHL with G-MPNN Yadati (2020), HC-Net Huang et al. (2024), ULTRA Galkin et al. (2024), and HYPER Huang et al. (2026), which run under the same fully inductive protocol as Huang et al. (2026). Table 7 summarizes arity handling, position use, and permutation behavior.

Table 7: Methods' ability to handle high-arity relations (**High-arity**), preserve positional information (**Pos. Info.**), invariance under legal intra-role permutations (**Legal** P), and ranking stability under illegal cross-role permutations (**Illegal** P). Illegal P refers to original-target MRR after illegal swaps, as in Figure 3(b): unordered models remain stable by ignoring all order, whereas SOHL remains stable by recovering role structure. RAM uses explicit roles; PosKHG uses absolute positions. Neither appears in Table 1.

Methods	Category	High-arity	Pos. Info.	Legal P	Illegal P
NaLP	Unordered	✓	✗	✓	✓
RAM	Role-aware	✓	✓	✓	✓
PosKHG	Role-aware	✓	✓	✗	✗
HypE, BoxE	Ordered	✓	✓	✗	✗
NBFNet, A*Net	Ordered	✗	✓	✗	✗
G-MPNN, HCNet	Ordered	✓	✓	✗	✗
ULTRA, KG-ICL	Ordered	✗	✓	✗	✗
HYPER	Ordered	✓	✓	✗	✗
(Ours) SOHL	Semi-ordered	✓	✓	✓	✓

NaLP Guan et al. (2019), RAM Liu et al. (2021), HypE Fatemi et al. (2019), BoxE Abboud et al. (2020), and PosKHG Chen et al. (2023) are discussed in Section 3 as unordered, role-aware, or ordered scoring models; they are not re-run on the 16 inductive splits.

The training mixtures used by ULTRA, HYPER, and SOHL are shown in Table 8.

Table 8: Training datasets used by different model variants.

Model	KHG				KG				
	JF	AUTO	WP	MFB	FB	WN	CoDeX	NELL	46G
ULTRA/HYPER/SOHL-3KG	✗	✗	✗	✗	✓	✓	✓	✗	✗
ULTRA/HYPER/SOHL-4KG	✗	✗	✗	✗	✓	✓	✓	✓	✗
ULTRA/HYPER/SOHL-50KG	✗	✗	✗	✗	✓	✓	✓	✓	✓
ULTRA/HYPER/SOHL-4HG	✓	✓	✓	✓	✗	✗	✗	✗	✗
ULTRA/HYPER/SOHL-3KG+2HG	✓	✗	✓	✗	✓	✓	✓	✗	✗
ULTRA/HYPER/SOHL-E2E Inference	Target training graph only								

The 3KG+2HG mix is the principal pretrained checkpoint used in the main experiments.

F Implementation Notes

All experiments are implemented in PyTorch and run on two NVIDIA RTX Pro 6000 GPUs. Table 9 lists the training and encoding settings used throughout the main experiments.

Table 9: Training and encoding settings used throughout the main experiments.

Setting	Value
Framework	PyTorch
Hardware	2× NVIDIA RTX Pro 6000
Seeds / reporting	4 seeds, mean ± std
Early stopping	5 epochs without validation improvement
Hyperedge encoder	pooling network Zaheer et al. (2017)
Role capacity M	k (one candidate anchor per position)
Gating factors	$\lambda = 0.5$, $\beta = 0.3$
Scoring head	MLP on $[\mathbf{h}_q^s; \mathbf{h}_v]$, trained with BCE
Pretraining / splits	HYPER Huang et al. (2026) mixtures and 16 inductive splits

Each reported MRR is the mean of four independent seeds, with early stopping after 5 epochs of no validation improvement. The hyperedge encoder in Equation 9 is a pooling network Zaheer et al. (2017). The role capacity is $M = k$, so each position is a candidate role anchor. Sensitivity on four families (Figure 4) selects gating factors $\lambda = 0.5$ and $\beta = 0.3$. The scoring head is an MLP over the concatenated role-aware query and candidate entity, trained with binary cross-entropy. Pretraining mixtures and inductive splits follow HYPER Huang et al. (2026). Anonymous code and data are at `https://anonymous.4open.science/r/SSH-1F4D`.