NeuSymEA: Neuro-symbolic Entity Alignment via Variational Inference

Shengyuan Chen
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
sheng-yuan.chen@polyu.edu.hk

Zheng Yuan
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
yzheng.yuan@connect.polyu.hk

Qinggang Zhang*
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
qinggangg.zhang@connect.polyu.hk

Wen Hua
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
wency.hua@polyu.edu.hk

Jiannong Cao
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
csjcao@comp.polyu.edu.hk

Xiao Huang
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
xiaohuang@comp.polyu.edu.hk

Abstract

Entity alignment (EA) aims to merge two knowledge graphs (KGs) by identifying equivalent entity pairs. Existing methods can be categorized into symbolic and neural models. Symbolic models, while precise, struggle with substructure heterogeneity and sparsity, whereas neural models, although effective, generally lack interpretability and cannot handle uncertainty. We propose NeuSymEA, a unified neuro-symbolic reasoning framework that combines the strengths of both methods to fully exploit the cross-KG structural pattern for robust entity alignment. NeuSymEA models the joint probability of all possible pairs' truth scores in a Markov random field, regulated by a set of rules, and optimizes it with the variational EM algorithm. In the E-step, a neural model parameterizes the truth score distributions and infers missing alignments. In the M-step, the rule weights are updated based on the observed and inferred alignments, handling uncertainty. We introduce an efficient symbolic inference engine driven by logic deduction, enabling reasoning with extended rule lengths. NeuSymEA achieves a significant 7.6% hit@1 improvement on $\text{DBP15K}_{\text{ZH-EN}}$ compared with strong baselines and demonstrates robustness in low-resource settings, achieving 73.7% hit@1 accuracy on $\text{DBP15K}_{\text{FR-EN}}$ with only 1% pairs as seed alignments. Codes are released at https://github.com/chensyCN/NeuSymEA-NeurIPS25.

1 Introduction

Knowledge graphs (KGs) are crucial for organizing structured knowledge about entities and their relationships, enhancing search capabilities across various applications. They are widely used in

*Correpsponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

question-answering systems [1, 2], recommendation systems [3, 4], SQL generation [5, 6], natural language processing [7–9], etc.. Despite their utility, real-world KGs often face issues like incompleteness, domain specificity, and language constraints, which hinder their effectiveness in cross-disciplinary or multilingual contexts [10, 11]. To address these issues, entity alignment (EA) aims to merge disparate KGs into a unified, comprehensive knowledge base by identifying and linking equivalent entities across different KGs. For example, aligning entities between a biomedical KG and a pharmaceutical KG allows for mining cross-discipline relationships through the aligned entities, such as identifying the same drugs and their effects on different diseases to enhance drug repurposing efforts. This alignment enables more nuanced exploration and interrogation of interconnected data, providing richer insights into how entities function across multiple domains.

Entity alignment models aim to determine the equivalence of two entities by assessing their alignment probability. Existing methods can be broadly categorized into symbolic models and neural models. Symbolic models [12–14] provide interpretable and precise inference by mining ground rules, but they struggle with aligning low-degree entities, especially those without aligned neighbors. In such cases, the lack of supporting rules leads to low recall. Conversely, neural models, such as translation models [15, 16] and graph convolutional networks (GCNs) [17–23], excel in recalling similar entities by embedding them in a continuous space, yet they often fail to distinguish entities with similar representations, causing a drop in precision as the entity pool grows. Neuro-symbolic models aim to combine the strengths of both approaches, offering robust reasoning ability for entity alignment in challenging scenarios.

However, neuro-symbolic reasoning in entity alignment (EA) faces several challenges. First, combining symbolic and neural models into a unified framework is suboptimal due to the difficulty in aligning their objectives. Current approaches either use neural models as auxiliary modules for symbolic models to measure entity similarity [14] or employ symbolic models to refine pseudo-labels [24, 25]. Second, in EA task, the search space for rules is large. Deriving ground rules from both intra-KG and inter-KG structural patterns results in an exponentially growing search space as rule length increases, complicating efficient rule weight estimation and inference. Finally, generating interpretations for EA remains underexplored. Effective interpretations should not only generate supporting rules but also quantify their confidence through rule weights.

To overcome these challenges, we propose NeuSymEA, a neuro-symbolic framework that combines the strengths of both symbolic and neural models. NeuSymEA models the joint probability of truth score assignment for all possible entity pairs using a Markov random field, regulated by a set of weighted rules. This joint probability is optimized via a variational EM algorithm. During the E-step, a neural model parameterizes the truth scores and infers the missing alignments. In the M-step, the rule weights are updated based on both observed and inferred alignments. To leverage long rules without suffering from the exponential search space, we employ logic deduction to decompose rules of any length into a set of unit-length sub-rules. This allows for efficient inference and weight updates for long rules. After training, the missing alignments are jointly inferred by both components. Additionally, we introduce an explainer to enhance interpretability. By reversing the rule decomposition process, we extract long rules as explicit supporting evidence for alignments and recover rule weights as quantified confidence scores. Our contributions are summarized as below:

- **A principled neuro-symbolic reasoning framework via variational EM:** While variational EM has been utilized in KG completion tasks [26, 27], adapting it directly to the EA task is nontrivial because they only consider single-KG structures. We bridge this gap by formulating truth scores and weighted cross-KG rules, and modeling the joint probability of the truth scores in a Markov random field regulated by the weighted cross-KG rules.
- **Efficient optimization via logical decomposition:** We introduce a logic deduction mechanism that decomposes long rules into shorter ones, significantly reducing the complexity of rule inference and enabling efficient reasoning over large knowledge graphs.
- **Interpretable inference:** The explainer utilizes learned rules to generate support paths for interpreting both aligned and misaligned pairs. It offers two modes: **(1) Hard-anchor mode**—generates supporting paths from prealigned anchor pairs; and **(2) Soft-anchor mode**—incorporates inferred anchor pairs for more informative interpretation.
- **Empirical validation and superior results:** NeuSymEA demonstrates state-of-the-art performance on benchmark datasets, delivering robust alignment accuracy and rule-based interpretations.

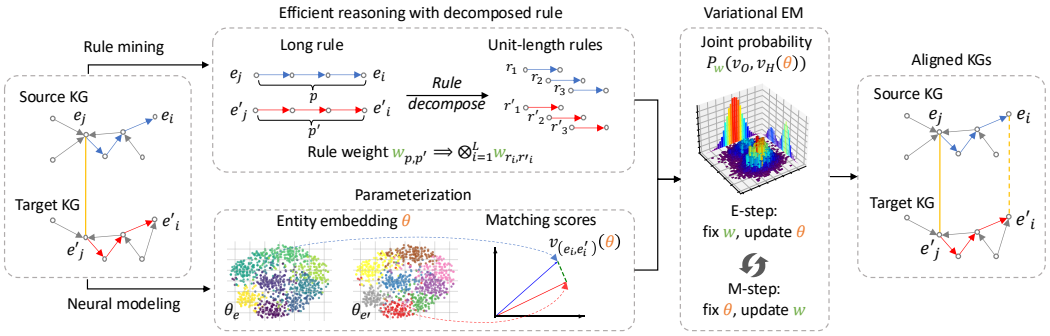

Figure 1: Framework illustration of NeuSymEA. The yellow solid line represents a pre-aligned entity pair. The symbolic model computes the path-level matching probability of entity pairs by mining rules with learned weights. The neural model learns embeddings and calculates entity-level matching scores. NeuSymEA models their agreement using a joint probability distribution over observed pairs and parameterized truth scores for hidden pairs, optimizing through a variational EM algorithm.

2 Preliminaries

2.1 Problem statement

A knowledge graph $\mathcal{G}$ comprises a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$, and a set of relation triples $\mathcal{T}$ where each triple $(e_i, r_k, e_j) \in \mathcal{T}$ represents a directional relationship between its head entity and tail entity. Given two KGs $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, $\mathcal{G}' = \{\mathcal{E}', \mathcal{R}', \mathcal{T}'\}$, and a set of observed aligned entity pairs $\mathcal{O} = \{(e_i, e'_i)|e_i \in \mathcal{E}, e'_i \in \mathcal{E}'\}_{i=1}^{n}$, the goal of entity alignment is to infer the missing alignments by reasoning with the existing alignments. This problem can be formulated in a probabilistic way: each pair $(e, e'), e \in \mathcal{E}, e' \in \mathcal{E}'$ is associated with a binary indicator variable $\boldsymbol{v}_{(e,e')}$. $\boldsymbol{v}_{(e,e')} = 1$ means (e, e') is an aligned pair, and $\boldsymbol{v}_{(e,e')} = 0$ otherwise. Given some observed alignments $\boldsymbol{v}_O = \{\boldsymbol{v}_{(e,e')} = 1\}_{(e,e')\in\mathcal{O}}$, we aim to predict the labels of the remaining hidden entity pairs $\mathcal{H} = \mathcal{E} \times \mathcal{E}' \backslash \mathcal{O}$, i.e., $\boldsymbol{v}_H = \{\boldsymbol{v}_{(e,e')}\}_{(e,e')\in\mathcal{H}}$.

2.2 Symbolic reasoning for entity alignment

Given an aligned pair (e_j, e'_j), a new aligned pair (e_i, e'_i) can be inferred with confidence score $w_{p,p'}$ if they are each connected to the existing pair via a relational path p and p' respectively, formally:

$$w_{p,p'} : (e_j \equiv e'_j) \wedge p(e_i, e_j) \wedge p'(e'_i, e'_j) \Longrightarrow (e_i \equiv e'_i), \tag{1}$$

where $p = |\mathcal{R}|^L, p' = |\mathcal{R}'|^L$ are a pair of paths each consisting of L connected relations, and $w_{p,p'}$ measures the rule quality that considers the intra-KG structure and inter-KG structure, such as the indicative of each path, and the similarity between two paths. By instantiating such *rule* with the constants (real entities and relations) in the KG pair, a symbolic model predicts the label distribution of an entity pair (e, e') by:

$$p_w(\boldsymbol{v}_{(e,e')}|\mathcal{G}, \mathcal{G}'), \quad \text{for } (e, e') \in \{\mathcal{O} \cup \mathcal{H}\}. \tag{2}$$

Using logic rules to infer the alignment probability can leverage the high-order structural information for effective alignment as well as provide interpretability. However, exact inference is intractable due to the massive amount of possible instantiated rules (exponential to L), limiting its applicability to real-world KGs.

3 Neuro-symbolic reasoning framework for entity alignment

3.1 Variational EM

Given a set of observed labels $\boldsymbol{v}_O$, our goal is to maximize the log-likelihood of these labels, i.e., $\log p_w(\boldsymbol{v}_O)$. Directly optimizing this objective is intractable because it requires computing an

integral over all the hidden variables. Instead, we optimize the evidence lower bound (ELBO) of the log-likelihood as follows:

$$p_w(\boldsymbol{v}_O) \geq E_{q(\boldsymbol{v}_H)} \left[\log p_w(\boldsymbol{v}_O, \boldsymbol{v}_H) - \log q(\boldsymbol{v}_H)\right] = \text{ELBO}(q, \boldsymbol{v}_O; w), \tag{3}$$

here, $q(\boldsymbol{v}_H)$ is a variational distribution of the hidden variables $\boldsymbol{v}_H$. This inequality holds for all q because $p_w(\boldsymbol{v}_O) = \text{ELBO}(q, \boldsymbol{v}_O; w) + D_{KL}(q(\boldsymbol{v}_H) \| p_w(\boldsymbol{v}_H \mid \boldsymbol{v}_O))$, where $D_{KL}(q(\boldsymbol{v}_H) \| p_w(\boldsymbol{v}_H \mid \boldsymbol{v}_O)) \geq 0$ is the KL-divergence between $q(\boldsymbol{v}_H)$ and $p_w(\boldsymbol{v}_H \mid \boldsymbol{v}_O)$. Under this framework, the log-likelihood $p_w(\boldsymbol{v}_O)$ can be optimized using an EM algorithm, an efficient method to find the maximum likelihood where the model depends on unobserved hidden variables: during the E-step, we fix w and update the variational distribution q; during the M-step, we update w to maximize the log-likelihood of all the entity pairs, i.e., $E_{q(\boldsymbol{v}_H)}[\log p_w(\boldsymbol{v}_O, \boldsymbol{v}_H)]$, as illustrated in Figure 1.

Explicitly representing the variational distribution q is parameter intensive, which requires $\approx |\mathcal{E}||\mathcal{E}'|$ variables because the observed pairs are very sparse. To this end, we parameterize q with a neural model as q_θ, with θ being the parameters of the neural model.

3.2 E-step: inference

In this step, we fix w and update q_θ to minimize the KL divergence D_{KL}. Directly minimizing the KL divergence is intractable, as it involves computing the entropy of q_θ. Therefore, we follow [26] and optimize the reverse KL divergence of q_θ and p_w, leading to the following objective:

$$\phi_{\boldsymbol{v}_H, \theta} = \sum_{(e,e') \in \mathcal{H}} \mathbf{E}_{p_w(\boldsymbol{v}_{(e,e')} | \boldsymbol{v}_O)} q_\theta(\boldsymbol{v}_H). \tag{4}$$

To optimize this objective, we first use the symbolic model with weighted rules to predict $p_w(\boldsymbol{v}_{(e,e')} \mid \boldsymbol{v}_O)$ for each $(e, e') \in \mathcal{H}$. If $p_w(\boldsymbol{v}_{(e,e')} \mid \boldsymbol{v}_O) > \delta$, where δ is a threshold, we treat this entity pair as a positive label; otherwise, we regard the pair as a negative pair that can be selected during negative sampling process of the neural model.

The observed labels can also be used as training data for supervised optimization. The objective is:

$$\phi_{\boldsymbol{v}_O, \theta} = \sum_{(e,e') \in \mathcal{O}} \log q_\theta(\boldsymbol{v}_{(e,e')} = 1). \tag{5}$$

The final objective for q_θ is obtained by combining these two objectives: $\phi_\theta = \phi_{\boldsymbol{v}_H, \theta} + \phi_{\boldsymbol{v}_O, \theta}$.

3.3 M-step: rule weight update

In this step, we fix q_θ and update the rule weight w to maximize $\text{ELBO}(q, \boldsymbol{v}_O; w)$. Since the right term of the ELBO in Equation 3 is constant when q_θ is fixed, the objective is equivalent to maximizing the left term $E_{q_\theta(\boldsymbol{v}_H)}[\log p_w(\boldsymbol{v}_O, \boldsymbol{v}_H)]$, which is the log-likelihood function.

Specifically, we start by predicting the labels of hidden variables using the current neural model. For each $(e, e') \in \mathcal{H}$, we predict the labels $\hat{\boldsymbol{v}}_{(e,e')}(\theta)$ and obtain the prediction set $\hat{\boldsymbol{v}}_H(\theta) = \{\hat{\boldsymbol{v}}_{(e,e')}(\theta)\}_{(e,e') \in \mathcal{H}}$. In this way, maximizing the likelihood practically becomes maximizing the following objective:

$$\phi_w = \log p_w(\boldsymbol{v}_O, \hat{\boldsymbol{v}}_H(\theta)). \tag{6}$$

To obtain the pseudo-label $\hat{\boldsymbol{v}}_{(e,e')}$ using q_θ, we employ the trained neural model to compute the matching score of any entity pair $(e, e') \in \mathcal{H}$. However, this strategy can easily introduce false positives into the pseudo-label set especially when the number of entities is large. To mitigate this, we consider one-to-one matching to sift only the most confident pairs. Practically, we first sort all pairs by their confidence score, then we annotate the pairs as positive following the order of the confidence. If a pair contains an entity observed in the annotated pairs, then this pair is skipped. This simple greedy strategy significantly reduces the amount of false positives.

4 Optimization and inference

4.1 Efficient optimization via logical deduction

Inference and learning with logic rules of length L can be intractable, as the search space for paths grows exponentially with increasing L. To enhance reasoning efficiency, we decompose a rule in

Equation 1 using logic deduction, inspired by [28] in KG completion:

$$w_{p,p'}: \quad (e_j \equiv e'_j) \wedge \left(\bigwedge_{k=1}^{L} r_k(e_{k-1}, e_k) \right) \wedge \left(\bigwedge_{k=1}^{L} r'_k(e'_{k-1}, e'_k) \right) \implies (e_i \equiv e'_i). \tag{7}$$

Here $\bigwedge_{k=1}^{L} r_k(e_{k-1}, e_k)$ represents the path formed by $r_1, r_2, ..., r_L$ connecting e_i to e_j with $e_0 = e_i$ and $e_k = e_j$. This long rule can be reorganized as *the combination of a series of unit-length logic reasoning*:

$$w_{p,p'}: \quad (e_j \equiv e'_j) \wedge \left(\bigwedge_{k=1}^{L} \left[r_k(e_{k-1}, e_k) \wedge r'_k(e'_{k-1}, e'_k) \right] \right) \implies (e_i \equiv e'_i). \tag{8}$$

In this way, each logic rule of length L can be viewed as *a deductive combination of L short rules of length 1.* At each step, following [12], we perform one-step inference to update $p_w(\boldsymbol{v}_{(e,e')})$ for each $(e, e') \in \mathcal{H}$ by aggregating the alignment probability from neighbors:

$$1 - \prod_{\substack{(e,r,e_t)\in\mathcal{T}, \\ (e',r',e'_t)\in\mathcal{T}'}} \left(1 - \eta(r) p_{sub}(r \subseteq r') p_w(\boldsymbol{v}_{(e_t,e'_t)})\right) \times \left(1 - \eta(r') p_{sub}(r' \subseteq r) p_w(\boldsymbol{v}_{(e_t,e'_t)})\right), \tag{9}$$

where $\eta(r)$ is a relation pattern of r measuring the uniqueness of e through relation r given a specified tail entity e_t, quantified by $\eta(r) = \frac{|\{e_t|(e_h,r,e_t)\in\mathcal{T}\}}{|\{(e_h,e_t)|(e_h,r,e_t)\in\mathcal{T})\}|}$. $p_{sub}(r \subseteq r')$ denotes the probability that relation r is a subrelation of r'. This technique enables inference with confidence by explicitly quantifying confidence w during each inference step by introducing η and $p_{sub}(r \subseteq r')$. Moreover, in this way, the update of the weight w simplifies to updating $p_{sub}(r \subseteq r')$ during the M-step (Equation 6), as $\eta(r)$ for each relation r is constant. In practice, the update of $p_{sub}(r \subseteq r')$ can be computed by:

$$\frac{\sum \left(1 - \prod_{(e'_h,r',e'_t)\in\mathcal{T}'} \left(1 - \boldsymbol{v}_{(e_h,e'_h)} \boldsymbol{v}_{(e_t,e'_t)}\right)\right)}{\sum \left(1 - \prod_{e'_h,e'_t\in\mathcal{E}'} \left(1 - \boldsymbol{v}_{(e_h,e'_h)} \boldsymbol{v}_{(e_t,e'_t)}\right)\right)}, \tag{10}$$

where $\boldsymbol{v}_{(e_h,e'_h)}$ and $\boldsymbol{v}_{(e_t,e'_t)}$ are labels (or pseudo-labels) from $\boldsymbol{v}_O \cup \hat{\boldsymbol{v}}_H(\theta)$.

After optimization, rule weights can be computed by taking the product of the importance scores η of relations and the sub-relation probabilities of the corresponding relation pair:

$$w_{p,p'} \coloneqq \prod_{k=1}^{L} \eta(r_k) \cdot \eta(r'_k) \cdot \frac{p_{sub}(r_k \subseteq r'_k) + p_{sub}(r'_k \subseteq r_k)}{2}. \tag{11}$$

In Appendix A.2, we provided a detailed complexity analysis, demonstrating that parameter complexity scales linearly with dataset size, while computational complexity is quadratic. Our implementation enables efficient execution through parallel computing and batch processing.

4.2 Inference with interpretability

To predict new alignments, there are two approaches: using the symbolic model or the neural model. The symbolic model infers alignment probabilities with the optimized weights w. Due to scalability concerns, symbolic methods generally adopt a lazy inference strategy that only preserves the confidently inferred pairs during inference. On the other hand, the neural model computes similarity scores for all entity pairs $(e, e') \in \mathcal{H}$ using the learned parameters θ, generating a ranked candidate list for each entity.

The evaluation of these models thus differs. Symbolic models are generally evaluated by precision, recall, and F1-score for their binary outputs, while neural models are assessed using hit@k and mean reciprocal ranks (MRR) for their ranked candidate lists. Following the practices in [14] and [24], we unify the evaluation metrics by treating the recall metric of symbolic models as equivalent to hit@1, facilitating comparison with neural models.

To enhance the interpretability of predictions, we adapt the optimized symbolic model into an explainer. For any given entity pair, the explainer generates a set of supporting rule path pairs that

justify their alignment, each associated with a confidence score indicating its significance. The explainer operates in two modes: ❶ hard-anchor mode, which generates supporting paths only from prealigned pairs, and ❷ soft-anchor mode, which includes paths from both prealigned and inferred pairs, providing more informative interpretations.

By integrating a breadth-first search algorithm (detailed in Appendix A.3), the explainer efficiently generates high-quality interpretations. For truly aligned pairs, it typically produces high-confidence interpretations, while for non-aligned pairs, the interpretations may result in an empty set (indicating no supporting evidence) or have low confidence scores. See Figure 4 for a visualized comparison.

5 Experiments

5.1 Experimental settings

Datasets. Main experiments use the DBP15K dataset, comprising three cross-lingual KG pairs: JA-EN, FR-EN, and ZH-EN. The original full version [29] of DBP15K resembles real-world KGs, posing challenges for GCN-based models due to sparsity and scale. Recent GCN-based models [19, 16, 17, 24] remove low-degree entities to get a smaller version with higher average degree. For thorough evaluation, we utilize **both full and condensed versions**. Dataset statistics are provided in Appendix B.1. For additional experiments on large KGs, we employ OpenEA [30] and DBP1M.

Two different dataset split strategies are used in the EA literature: ❶ a 3:7 train/test split, and ❷ a 5-fold cross-validation scheme with a 2:1:7 ratio for training, validation, and test sets, as used in OpenEA [30]. We adopt the latter for all algorithms to ensure fair comparison.

Baselines and metrics. Baseline models include seven neural models – GCNAlign [19], AlignE, BootEA [16], RREA [17], Dual-AMN [18], LightEA [20], PEEA [31], one symbolic models – PARIS [12], and two neuro-symbolic models – PRASE [14], EMEA [24]. We use Hit@1, Hit@10, and MRR as the evaluation metrics. For PARIS and PRASE that have binary outputs, we report their recall as Hit@1, following [24]. For RREA, Dual-AMN, and LightEA, which offer both basic and iterative versions, we adopt the iterative ones due to their generally superior performance.

Hyperparameters. NeuSymEA involves two main hyperparameters: the number of EM iterations and the symbolic model's threshold δ for selecting positive pairs. We tune them on the validation set, searching δ in $0.6, 0.7, 0.8, 0.9, 0.95, 0.98, 0.99$ and the number of iterations from 1 to 9. As shown in Section 5.3, NeuSymEA is robust to δ and converges quickly.

5.2 Results

5.2.1 Comparison with baselines

Table 1 compares NeuSymEA with baseline models on two versions of the DBP15K dataset: the full and the condensed version. Results for PRASE and EMEA on the condensed DBP15K are sourced from the original EMEA paper. The results yield three key observations:

First, NeuSymEA surpasses both symbolic and neural models. By integrating symbolic reasoning with neural representations, it: 1) captures multi-hop relational structures across KGs using rules; and 2) learns effective entity representations to compute pairwise similarities. **Second, NeuSymEA outperforms other neuro-symbolic models.** This improvement can be largely attributed to the model objective design in our framework. While PRASE and EMEA treat the symbolic and neural models as separate components, NeuSymEA unifies them under a joint probability objective. This enables joint optimization via Variational EM, yielding a more coherent and convergent solution with superior performance. **Finally, NeuSymEA demonstrates robustness across both full and condensed datasets.** Comparisons between two groups of results offer an interesting insight: neural models experience significant performance degradation in the full version of DBP15K (e.g., MRR of Dual-AMN decreases from 0.815 to 0.717 on JA-EN), while symbolic models, in contrast, show improvements. We attribute this to their different matching mechanisms: (1) neural models rely on *entity-level matching*, which is sensitive to dataset size. The full DBP15K includes more low-degree entities, increasing similar embeddings and reducing precision. (2) Symbolic models use *path-level matching*, which is less affected by dataset size but vulnerable to substructure heterogeneity. The full DBP15K's additional connections via long-tail entities enhance rule-mining, boosting symbolic

Table 1: Entity alignment results on DBP15K dataset. The suffixes "-D" and "-L" indicate the use of Dual-AMN and LightEA as the neural models. The results of RREA and EMEA are omitted on the full dataset due to an OOM (Out of Memory) error.

Category	Model	JA-EN			FR-EN			ZH-EN		
		Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
Results on the full DBP15K dataset										
Neural	GCNAlign	0.221	0.461	0.302	0.205	0.475	0.295	0.189	0.438	0.271
	BootEA	0.454	0.782	0.564	0.443	0.799	0.564	0.486	0.814	0.600
	AlignE	0.356	0.715	0.476	0.346	0.731	0.475	0.333	0.690	0.453
	Dual-AMN	0.627	0.883	0.717	0.652	0.908	0.744	0.650	0.884	0.732
	LightEA	0.736	0.894	0.793	0.782	0.919	0.832	0.725	0.874	0.779
Symbolic	PARIS	0.589	-	-	0.618	-	-	0.603	-	-
Neuro-symbolic	PRASE	0.611	-	-	0.647	-	-	0.652	-	-
Ours	NeuSymEA-D	**0.806**	**0.942**	**0.855**	0.827	**0.952**	**0.871**	**0.801**	**0.925**	**0.843**
	NeuSymEA-L	0.781	0.907	0.826	**0.834**	0.937	**0.871**	0.785	0.894	0.825
Results on the condensed DBP15K dataset										
Neural	GCNAlign	0.331	0.662	0.443	0.325	0.688	0.446	0.335	0.653	0.442
	BootEA	0.530	0.829	0.631	0.579	0.872	0.961	0.575	0.847	0.668
	AlignE	0.433	0.783	0.552	0.457	0.821	0.580	0.474	0.806	0.587
	RREA	0.749	**0.935**	0.818	0.797	**0.958**	0.859	0.762	**0.938**	0.827
	Dual-AMN	0.750	0.927	0.815	0.793	0.954	0.854	0.756	0.919	0.816
	LightEA	0.778	0.911	0.828	0.827	0.943	0.830	0.770	0.894	0.816
	PEEA	0.703	0.912	0.777	0.748	0.937	0.815	0.726	0.905	0.790
Symbolic	PARIS	0.565	-	-	0.584	-	-	0.543	-	-
Neuro-symbolic	PRASE	0.580	-	-	0.622	-	-	0.593	-	-
	EMEA	0.736	-	0.807	0.773	-	0.841	0.748	-	0.815
Ours	NeuSymEA-D	0.805	0.930	0.849	0.835	0.953	0.879	**0.815**	0.926	**0.855**
	NeuSymEA-L	**0.811**	0.928	**0.854**	**0.858**	0.954	**0.894**	0.804	0.904	0.840

model performance. By integrating symbolic reasoning with KG embeddings, NeuSymEA overcomes these limitations, ensuring robustness to variations in dataset scale and structure.

5.2.2 Evolution of rules and embeddings

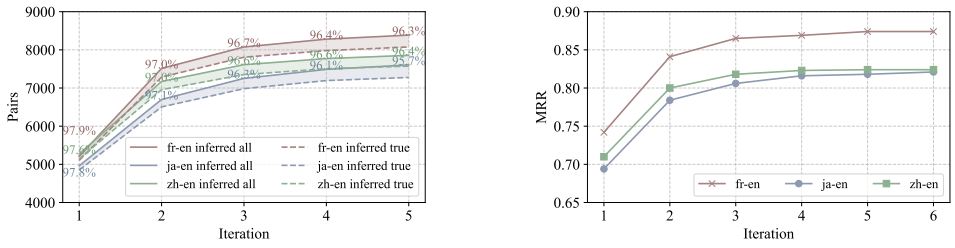

Figure 2: (Left) Evolution of rule inferred pairs, with solid lines representing total inferred pairs and dashed lines representing true inferred. The shaded areas indicate the number of false pairs. Precision values are annotated at each data point. (Right) Convergence of MRR of the neural model.

We study how rules and embeddings evolve and interact with each other during the EM steps, with results shown in Figure 2. Results in the left subplot indicate that in each EM iteration, the number of rule-inferred pairs grows consistently with high precision, implying that the embedding model continuously improves the inference performance of rules. These precise pairs, in turn, enhance the performance of the neural model. As shown in the right subfigure, the MRR of the neural model converges within a few iterations.

5.2.3 Scalability on large datasets

Figure 3 illustrate the scalability of NeuSymEA. We evaluated its hit@1 accuracy and runtime efficiency across datasets of varying entity sizes, namely DBP15K, OpenEA100K, and DBP1M.

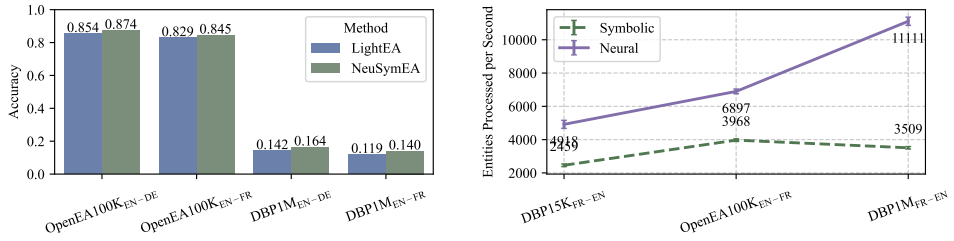

Figure 3: Scalability analysis on large scale KGs. (Left) Hit@1 alignment performance on large KGs. (Right) Per-second processed entities of neural and symbolic components on different scales of KGs.

Performance scalability. In the left subfigure, we compare NeuSymEA with LightEA - the only strong baseline capable of processing large-scale knowledge graphs with million-scale entities. The results demonstrate NeuSymEA's superior scalability in terms of performance.

Runtime Scalability. We separately evaluate the runtime performance of NeuSymEA's neural and symbolic reasoning components. For the neural component, efficiency (entities processed per second) increases with dataset size. We attribute this to higher GPU utilization on larger datasets, which enhances computational efficiency. Conversely, the symbolic component's efficiency initially increases but then slightly decreases. Upon investigation, we found that this phenomenon is related to the multiprocessing and batch-processing implementation. For smaller datasets (e.g., DBP15K), the overhead from process initialization and termination is significant. As dataset size grows to OpenEA100K, this overhead becomes negligible relative to inference runtime, leading to an efficiency improvement. When moving from OpenEA100K to DBP1M, the per-second processed entities slightly decline as the quadratic complexity of inference computation dominates.

5.2.4 Interpretations by the explainer

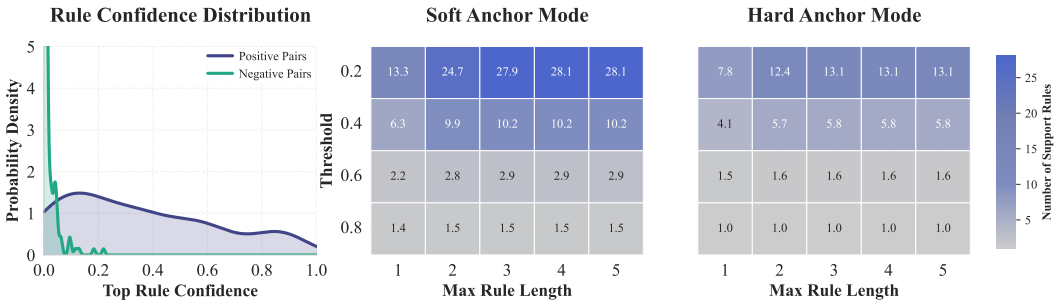

Figure 4: (Left) Probability density of the top supporting rule's confidence; (Middle) Number of supporting rules relative to the maximum rule lengths under the soft anchor mode; (Right) Number of supporting rules relative to the maximum rule lengths under the hard anchor mode.

We investigate the interpretations generated by the explainer on the FR-EN dataset in Figure 4. The left subfigure displays the probability density of confidence scores for supporting rules associated with entity pairs. Positive pairs come from the test set, while negative pairs are generated by randomly replacing one entity in each positive pair. The distinct confidence distributions show that positive pairs generally have stronger alignment evidence, as expected. Upon further examination, we found that many test pairs are isolated, i.e., they lack directly aligned neighbors. Despite this, NeuSymEA successfully generates supporting rules for isolated pairs by exploiting multi-hop dependencies.

The middle and right subfigures examine the impact of rule length on the explainer's effectiveness, showing the number of supporting rules for positive pairs as the maximum rule length increases. In soft anchor mode, the explainer produces more high-quality supporting rules than in hard anchor mode by using inferred pairs as complementary anchors, mitigating substructure sparsity. While longer maximum rule lengths yield more high-quality supporting rules, however, the long rules have a lower confidence distribution compared to short rules. This can be attributed to our method for calculating confidence: the logical deduction-based approach computes a rule's confidence as the

product of the confidences of its decomposed unit-length sub rules (as in Equation 11). For instance, a rule comprising two unit-length sub-rules, each with a confidence of 0.8, has a combined confidence of $0.8 \times 0.8 = 0.64$. Thus, confidence scores typically decrease as rule length increases.

Table 2: Examples of supporting rules for query pairs in FR-EN. Anchor pairs are shown in bold.

Query Pair	Supporting Rule	Confidence
Maison_de_Savoie House_of_Savoy	(Humbert_II_(roi_d'Italie), dynastie, Maison_de_Savoie), (Humbert_II_(roi_d'Italie), conjoint, **Marie-José_de_Belgique**) (Umberto_II_of_Italy, house, House_of_Savoy), (Umberto_II_of_Italy, spouse, **Marie_José_of_Belgium**)	0.80
Légion_espagnole Spanish_Legion	(Légion_espagnole, commandantHistorique, Francisco_Franco), (Francisco_Franco, conjoint, **Carmen_Polo**) (Spanish_Legion, notableCommanders, Francisco_Franco), (Francisco_Franco, spouse, **Carmen_Polo,_1st_Lady_of_Meirás**)	0.59
Premier_ministre_du_Danemark Prime_Minister_of_Denmark	(Premier_ministre_du_Danemark, titulaireActuel, **Lars_Løkke_Rasmussen**) (Prime_Minister_of_Denmark, incumbent, **Lars_Løkke_Rasmussen**)	0.79

5.2.5 Robustness in low resource scenario

Table 3: Comprehensive results with different ratios of training data, with best shown in bold.

Dataset	Model	1%			5%			10%			20%		
		H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
JA-EN	AlignE	0.007	0.034	0.016	0.080	0.268	0.143	0.244	0.588	0.356	0.433	0.783	0.552
	BootEA	0.010	0.040	0.021	0.379	0.683	0.481	0.468	0.779	0.573	0.530	0.829	0.631
	GCNAlign	0.029	0.128	0.063	0.127	0.368	0.206	0.209	0.515	0.310	0.331	0.662	0.443
	PARIS	0.145	-	-	0.340	-	-	0.450	-	-	0.565	-	-
	PRASE	0.163	-	-	0.432	-	-	0.508	-	-	0.580	-	-
	Dual-AMN	0.239	0.519	0.334	0.509	0.795	0.611	0.652	0.887	0.738	0.750	0.927	0.815
	RREA	0.253	0.486	0.332	0.558	0.830	0.653	0.672	**0.903**	0.756	0.789	**0.956**	0.853
	LightEA	0.291	0.514	0.363	0.627	0.806	0.689	0.714	0.874	0.771	0.778	0.911	0.828
	EMEA	0.411	-	0.488	0.630	-	0.710	0.688	-	0.764	0.736	-	0.807
	PEEA	0.242	0.519	0.333	0.490	0.785	0.589	0.612	0.834	0.679	0.703	0.912	0.777
	NeuSymEA-D	0.481	0.684	0.550	0.692	0.855	0.749	0.742	0.895	0.796	0.835	0.953	0.879
	NeuSymEA-L	**0.632**	**0.779**	**0.683**	**0.733**	**0.870**	**0.781**	**0.773**	0.900	**0.818**	**0.858**	0.954	**0.894**
FR-EN	AlignE	0.008	0.040	0.019	0.127	0.408	0.217	0.347	0.733	0.475	0.457	0.821	0.580
	BootEA	0.009	0.041	0.020	0.418	0.746	0.529	0.490	0.809	0.598	0.579	0.872	0.681
	GCNAlign	0.027	0.119	0.058	0.133	0.388	0.215	0.215	0.539	0.321	0.325	0.688	0.446
	PARIS	0.195	-	-	0.401	-	-	0.479	-	-	0.584	-	-
	PRASE	0.227	-	-	0.514	-	-	0.575	-	-	0.633	-	-
	Dual-AMN	0.293	0.631	0.407	0.598	0.886	0.703	0.717	0.928	0.797	0.793	0.954	0.854
	RREA	0.289	0.583	0.389	0.628	0.895	0.725	0.717	0.932	0.796	0.789	**0.956**	0.853
	LightEA	0.430	0.663	0.509	0.723	0.885	0.781	0.779	0.914	0.828	0.827	0.943	0.870
	EMEA	0.480	-	0.565	0.677	-	0.757	0.727	-	0.802	0.773	-	0.841
	PEEA	0.285	0.588	0.385	0.552	0.812	0.642	0.665	0.875	0.738	0.748	0.937	0.815
	NeuSymEA-D	0.642	0.833	0.709	0.768	0.916	0.820	0.811	**0.939**	0.856	0.835	0.953	0.879
	NeuSymEA-L	**0.737**	**0.874**	**0.785**	**0.806**	**0.921**	**0.848**	**0.827**	0.937	**0.867**	**0.858**	0.954	**0.894**
ZH-EN	AlignE	0.006	0.033	0.016	0.127	0.368	0.206	0.296	0.635	0.407	0.474	0.806	0.587
	BootEA	0.006	0.029	0.014	0.396	0.689	0.495	0.498	0.782	0.594	0.575	0.847	0.668
	GCNAlign	0.041	0.155	0.080	0.147	0.396	0.229	0.225	0.519	0.323	0.335	0.653	0.442
	PARIS	0.059	-	-	0.333	-	-	0.429	-	-	0.543	-	-
	PRASE	0.241	-	-	0.461	-	-	0.522	-	-	0.593	-	-
	Dual-AMN	0.375	0.666	0.480	0.582	0.830	0.672	0.676	0.892	0.755	0.756	0.918	0.816
	RREA	0.316	0.564	0.403	0.605	**0.858**	0.696	0.686	**0.901**	0.765	0.760	**0.934**	0.823
	LightEA	0.507	0.673	0.565	0.670	0.819	0.723	0.727	0.860	0.775	0.770	0.894	0.816
	EMEA	0.517	-	0.591	0.665	-	0.738	0.706	-	0.777	0.748	-	0.815
	PEEA	0.288	0.586	0.388	0.532	0.801	0.622	0.649	0.871	0.710	0.725	0.905	0.790
	NeuSymEA-D	0.589	0.750	0.645	0.704	0.856	0.757	0.763	0.897	0.809	**0.815**	0.926	**0.855**
	NeuSymEA-L	**0.676**	**0.799**	**0.720**	**0.735**	**0.858**	**0.779**	**0.773**	0.882	**0.811**	0.804	**0.934**	0.841

Table 3 demonstrates the model performance under low-resource settings. As the percentage of training data decreases, all models experience noticeable drops in Hit@1 performance. Despite this, NeuSymEA exhibits remarkable robustness across all datasets, consistently outperforming other models. Notably, with only 1% of pairs used as training data, NeuSymEA achieves a Hit@1 score exceeding 0.7 on FR-EN, rivaling or even surpassing the performance of some state-of-the-art models trained on 20% of the data.

5.3 Parameter analysis

We present the hit@1 performance of NeuSymEA across three datasets, varying hyperparameters, illustrated by a three-dimensional graph. The threshold hyperparameter δ is explored within the set {0.6, 0.7, 0.8, 0.9, 0.95, 0.98, 0.99}, while the number of EM iterations ranges from 1 to 9. Performance levels are indicated using a colormap. Performance sensitivity analysis in Figure 5

reveals that for all datasets, performance generally improves as the iteration increases. On the other hand, the performance is less sensitive to the threshold δ.

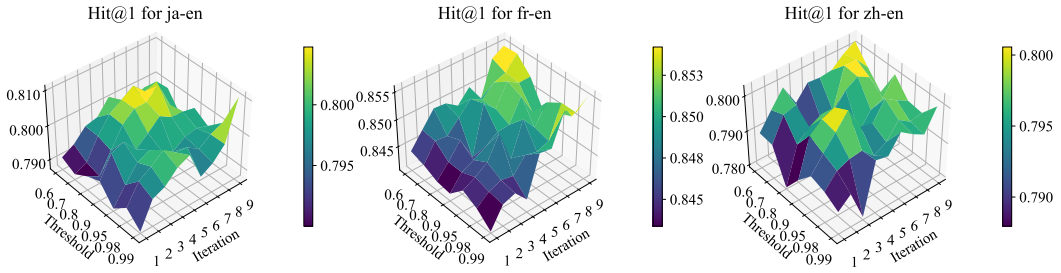

Figure 5: Performance sensitivity to hyperparameters iteration and threshold δ.

6 Related work

Neuro-symbolic reasoning on knowledge graphs. Neuro-symbolic methods aim to combine symbolic reasoning with neural representation learning, leveraging the precision and interpretability of symbolic approaches alongside the scalability and high recall of neural methods. In KG completion task, [32] and [33] employ horn rules to regularize the learning of KG embeddings; [34] and [28] model the rule-based predictions as distributions conditioned on the input relational sequences, and parameterize these distributions using a recurrent neural network; [26], [27] and [35] models the joint probability of the neural model and the symbolic model with a Markov random field, and employ gradient descent for weight updates. Despite extensive advancements of neuro-symbolic reasoning in KG completion, these studies only consider single-KG structures, thus cannot be directly adopted to entity alignment which requires consideration of inter-KG structures.

Entity alignment. Recent models have sought to combine symbolic and neural approaches for entity alignment. For instance, [14] enhances probabilistic reasoning with KG embeddings to measure entity-level and relation-level similarities. [24] implements self-bootstrapping with pseudo-labeling in a neural framework, using rules to choose confident pseudo-labels. However, it relies solely on unit-length rules, which restricts its effectiveness for long-tail entities. In contrast, we employ logic deduction to scale symbolic reasoning with long rules of any length. Unlike existing work that separately handles two reasoning components, our work models the unified joint probability of symbolic and neural inference within a markov random field. The symbolic component captures intra-KG and inter-KG structures using weighted logic rules, while the neural model learn expressive patterns in the embedding space.

7 Limitations

NeuSymEA is currently designed for EA between two KGs. Extending it to align multiple KGs simultaneously may require iterative pairwise alignments, which could be inefficient. A more sophisticated optimization paradigm is needed to adapt NeuSymEA for scalable multi-KG alignment.

8 Conclusions

We presented NeuSymEA, a unified and extensible neuro-symbolic framework for entity alignment. By unifying neural and symbolic reasoning, NeuSymEA addresses the challenges of substructure heterogeneity, sparsity, and uncertainty in real-world KGs. Empirical results demonstrate NeuSymEA's clear improvements over baselines and robustness under limited resources. By delivering interpretable alignment predictions with uncertainty scores, NeuSymEA advances knowledge fusion, enabling effective and trustworthy entity alignment.

Acknowledgement

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 25208322).

References

[1] Hannah Bast and Elmar Haussmann. More accurate question answering on freebase. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1431–1440, 2015.

[2] Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. Hierarchy-aware multi-hop question answering over knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, page 2519–2527, 2023.

[3] Rose Catherine and William Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM conference on recommender systems*, pages 325–332, 2016.

[4] Yuanchen Bei, Hao Chen, Shengyuan Chen, Xiao Huang, Sheng Zhou, and Feiran Huang. Non-recursive cluster-scale graph interacted model for click-through rate prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3748–3752, 2023.

[5] Qinggang Zhang, Hao Chen, Junnan Dong, Shengyuan Chen, Feiran Huang, and Xiao Huang. Structure-guided large language models for text-to-SQL generation. In *Forty-second International Conference on Machine Learning*, 2025.

[6] Zheng Yuan, Hao Chen, Zijin Hong, Qinggang Zhang, Feiran Huang, Qing Li, and Xiao Huang. Knapsack optimization-based schema linking for llm-based text-to-sql generation. *arXiv preprint arXiv:2502.12911*, 2025.

[7] Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Yi Chang, et al. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*, 2025.

[8] Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *arXiv preprint arXiv:2506.05690*, 2025.

[9] Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. You don't need pre-built graphs for rag: Retrieval augmented generation with adaptive reasoning structures. *arXiv preprint arXiv:2508.06105*, 2025.

[10] Qinggang Zhang, Keyu Duan, Junnan Dong, Pai Zheng, and Xiao Huang. Logical reasoning with relation network for inductive knowledge graph completion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4268–4277, 2024.

[11] Qinggang Zhang, Junnan Dong, Keyu Duan, Xiao Huang, Yezi Liu, and Linchuan Xu. Contrastive knowledge graph error detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2590–2599, 2022.

[12] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. In *Proceedings of the 38th International Conference on Very Large Databases*, pages 157–168, 2012.

[13] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *Proceedings of the 10th International Semantic Web Confernece*, pages 273–288, 2011.

[14] Zhiyuan Qi, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Yuejia Xiang, Ningyu Zhang, and Yefeng Zheng. Unsupervised knowledge graph alignment by probabilistic reasoning and semantic embedding. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[15] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, page 1511–1517, 2017.

[16] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, number 2018, 2018.

[17] Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. Relational reflection entity alignment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1095–1104, 2020.

[18] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. Boosting the speed of entity alignment 10×: Dual attention matching network with normalized hard sample mining. In *Proceedings of the Web Conference 2021*, pages 821–832, 2021.

[19] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 349–357, 2018.

[20] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. LightEA: A scalable, robust, and interpretable entity alignment framework via three-view label propagation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[21] Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. Taming language models for text-attributed graph learning with decoupled aggregation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3463–3474, 2025.

[22] Zhixun Li, Xin Sun, Yifan Luo, Yanqiao Zhu, Dingshuo Chen, Yingtao Luo, Xiangxin Zhou, Qiang Liu, Shu Wu, Liang Wang, et al. GSLB: the graph structure learning benchmark. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Zirui Liu, Chen Shengyuan, Kaixiong Zhou, Daochen Zha, Xiao Huang, and Xia Hu. Rsc: accelerate graph neural networks training via randomized sparse computations. In *International Conference on Machine Learning*, pages 21951–21968. PMLR, 2023.

[24] Bing Liu, Harrisen Scells, Wen Hua, Guido Zuccon, Genghong Zhao, and Xia Zhang. Guiding neural entity alignment with compatibility. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 491–504. Association for Computational Linguistics, 2022.

[25] Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. Entity alignment with noisy annotations from large language models. *arXiv preprint arXiv:2405.16806*, 2024.

[26] Meng Qu and Jian Tang. Probabilistic logic neural networks for reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[27] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient probabilistic logic reasoning with graph neural networks. In *International Conference on Learning Representations*, 2020.

[28] Keiwei Cheng, Nesreen K Amed, and Yizhou Sun. Neural compositional rule learning for knowledge graph reasoning. In *International Conference on Learning Representations*, 2023.

[29] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of the 16th International Semantic Web Confernece*, pages 628–644. Springer, 2017.

[30] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11):2326–2340, 2020.

[31] Wei Tang, Fenglong Su, Haifeng Sun, Qi Qi, Jingyu Wang, Shimin Tao, and Hao Yang. Weakly supervised entity alignment with positional inspiration. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 814–822, 2023.

[32] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 192–202, 2016.

[33] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[34] Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 179–189, 2022.

[35] Shengyuan Chen, Yunfeng Cai, Huang Fang, Xiao Huang, and Mingming Sun. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

A Notations and algorithms

A.1 Notations

Table 4: Notations

Notation	Description
$\mathcal{G}, \mathcal{G}'$	The source and target knowledge graphs, respectively
$\mathcal{E}, \mathcal{E}'$	The sets of entities in $\mathcal{G}$ and $\mathcal{G}'$, respectively
$\mathcal{R}, \mathcal{R}'$	The sets of relations in $\mathcal{G}$ and $\mathcal{G}'$, respectively
$\mathcal{T}, \mathcal{T}'$	The sets of relational triplets in $\mathcal{G}$ and $\mathcal{G}'$, respectively
$\mathcal{O}$	The set of observed aligned entity pairs between two knowledge graphs $\mathcal{G}$ and $\mathcal{G}'$
$\mathcal{H}$	Set of unobserved entity pairs, i.e., $\mathcal{E} \times \mathcal{E}' \backslash \mathcal{O}$
$\boldsymbol{v}_{(e,e')}$	Binary indicator variable for an entity pair (e, e'), where $\boldsymbol{v}_{(e,e')} = 1$ indicates alignment
$w_{p,p'}$	Confidence score of a rule-inferred alignment based on paths p and p'
$p_w(\boldsymbol{v}_{(e,e')} \mid \mathcal{G}, \mathcal{G}')$	Probability distribution of the alignment indicator $\boldsymbol{v}_{(e,e')}$ given knowledge graphs $\mathcal{G}$ and $\mathcal{G}'$
θ	Parameters of the neural model
δ	Threshold to select positive pair from the symbolic model
$\eta(r)$	Relation pattern measuring the uniqueness of an entity through relation r

A.2 Complexity analysis of the symbolic reasoning

In the following, we present the analysis of runtime complexity and parameter complexity one by one.

A.2.1 Runtime complexity

In variational inference, the process of learning and inferring long rules (Equation 7) is simplified by decomposing them into unit-length rules (Equation 8). Consequently, rule weight learning (Equation 10) is only conducted for unit-length rules. The inference process for an L-length rule is then estimated by iteratively applying inference steps with unit-length rules (Equation 9) for L iterations. This strategy effectively avoids the exponential search space associated with longer rules, making the computational complexity of the inference linear with respect to the rule length L.

Each iteration of reasoning with unit-length rules comprises an inference step (Equation 9) and a rule-weight learning step (Equation 10). These steps require computing the matching probability for all possible entity pairs and relation pairs, respectively. As a result, the computational complexity of the inference step and the weight updating step are $O(|\mathcal{E}||\mathcal{E}'|)$ and $O(|\mathcal{R}||\mathcal{R}'|)$, respectively.

Thus, the total **computational complexity** for reasoning with an L-length rule is $O(L \cdot (|\mathcal{E}||\mathcal{E}'| + |\mathcal{R}||\mathcal{R}'|))$. Given that entity sizes are typically much larger than relation sizes, this complexity can be approximated as $O(L \cdot |\mathcal{E}||\mathcal{E}'|)$.

Notably, the computations involved in Equation 9 and Equation 10 can be accelerated through parallel processing, which we have implemented. This optimization reduces the **runtime complexity** to $O\left(L \cdot \frac{|\mathcal{E}||\mathcal{E}'|}{n}\right)$, where n represents the number of CPU cores available for parallelization.

A.2.2 Parameter complexity

The total number of alignment probabilities for all entity pairs is $|\mathcal{E}||\mathcal{E}'|$, which is large when the entity sizes increase. We adopt a lazy inference strategy to enhance parameter efficiency. This strategy involves only saving the alignment probabilities of the most probable alignments:

$$\left\{ p_w(v_{(e_i,e_i')}), |, e_i \in \mathcal{E}, e_i' \in \mathcal{E}', p_w(v_{(e_i,e_i')}) = \max\left(\max_{e \in \mathcal{E}} p_w(v_{(e,e_i')}), \max_{e' \in \mathcal{E}'} p_w(v_{(e_i,e')}) \right) \right\} \quad (12)$$

Probabilities of other entity pairs can be inferred from these saved alignment probabilities using Equation 9. In this way, **parameter complexity** is reduced to $O(\max(|\mathcal{E}| + |\mathcal{E}'|))$.

A.3 Pseudo-code of Explainer

Below is the pseudo-code of how the explainer generates supporting rules as interpretations for the query pair. It consists of two stages: searching reachable anchor pairs, and parsing rule paths as well as calculating rule confidences.

Algorithm 1 Generating Interpretations for the Queried Entity Pair with Weighted Rules

Inputs: Subrelation probabilities $p_{sub}(r \subseteq r'), p_{sub}(r' \subseteq r)$ for $r, r' \in \mathcal{R}$; Knowledge Graph pair $(\mathcal{G}, \mathcal{G}')$; Maximum rule length $\mathcal{L}$; Anchor pairs $\mathcal{A}$ with source-to-target mapping `S2T` and target-to-source mapping `T2S`; Query entity pair (e_q, e_q')
Outputs: Ranked rules based on confidence
1. Search Reachable Anchor Pairs within Max Depth $\mathcal{L}$
$RN \leftarrow \text{BFS}(e_q, \mathcal{G}, \mathcal{L})$ /* Search reachable neighbors of e_q using breadth-first search, max depth $\mathcal{L}$ */
$RN' \leftarrow \text{BFS}(e_q', \mathcal{G}', \mathcal{L})$ /* Search reachable neighbors of e_q' using breadth-first search, max depth $\mathcal{L}$ */
$RN_a \leftarrow RN \cup \texttt{T2S}(RN'; \mathcal{A})$ /* Find source nodes of reachable anchor pairs using hash mapping */
$RA \leftarrow \{(e, \texttt{S2T}(e; \mathcal{A})) \mid e \in RN_a\}$ /* Identify reachable anchor pairs */
2. Parse and Rank Rules Based on Confidence
for $\forall (e, e') \in RA$ **do**
  Extract paths: $p(e, e_q) = r_1 \wedge r_2 \wedge \ldots, p'(e', e_q') = r_1' \wedge r_2' \wedge \ldots$
  if $|p(e, e_q)| \neq |p'(e', e_q')|$ **then**
    $w_{p(e,e_q),p'(e',e_q')} \leftarrow 0$ /* If path lengths don't match, rule confidence is 0 */
  else
    $w_{p(e,e_q),p'(e',e_q')} \leftarrow \prod_{i=1}^{|p|} \eta(r_i) \cdot \eta(r_i') \cdot \frac{p_{sub}(r_i \subseteq r_i') + p_{sub}(r_i' \subseteq r_i)}{2}$ /* Compute rule confidence by products of subrelation probabilities and relation functionalities */
  end if
end for
Sort the rules (p, p') by $w_{p,p'}$ in descending order
Return the ranked rules

B Experimental details

B.1 Dataset statistics

The DBP15K dataset, designed for cross-lingual knowledge graph alignment, has two versions: full and condensed. The original full version resembles real-world knowledge graphs, including comprehensive data across three language pairs with many sparsely connected low-degree entities. The condensed version, derived by JAPE and adopted by later methods (GCNAlign, RREA, Dual-AMN, LightEA), removes these low-degree entities and their connected triples to create a smaller (and higher average degree) dataset suitable for GCN-based methods. Detailed information about the two dataset versions can be found in the "dataset" section of readme in JAPE's official implementation[2].

The dataset statistics of them are shown in Table 5 and Table 6. In this paper, we adopt both versions of DBP15K for comprehensive evaluation. Specifically, the full dataset is sparser and larger in scale in scale due to the inclusion of low-degree entities, thus suitable for evaluating the models' robustness to sparsity and large scale.

Table 5: Data statistics of the full DBP15K dataset.

Datasets	KG	Entities	Relations	Rel. Triplets	Aligned Entity Pairs
ZH-EN	Chinese (zh)	66,469	2,830	153,929	15,000
	English (en)	98,125	2,317	237,674	
JA-EN	Japanese (ja)	65,744	2,043	164,373	15,000
	English (en)	95,680	2,096	233,319	
FR-EN	French (fr)	66,858	1,379	192,191	15,000
	English (en)	105,889	2,209	278,590	

[2]https://github.com/nju-websoft/JAPE

Table 6: Data statistics of the condensed DBP15K dataset.

Datasets	KG	Entities	Relations	Rel. Triplets	Aligned Entity Pairs
ZH-EN	Chinese (zh)	19,388	1,701	70,414	15,000
	English (en)	19,572	1,323	95,142	
JA-EN	Japanese (ja)	19,814	1,299	77,214	15,000
	English (en)	19,780	1,153	93,484	
FR-EN	French (fr)	19,661	903	105,998	15,000
	English (en)	19,993	1,208	115,722	

B.2 Efficiency analysis

To provide a comprehensive understanding of the computational efficiency of our model, we report the runtime and memory usage during the experiments. The results, as summarized in Table Table 7, demonstrate that our model achieves efficient performance with a runtime of 15 minutes, a memory consumption of 868 MB, and a GPU memory usage of 4.33 GB. These metrics highlight the practicality of our approach in terms of resource utilization. Hardware configurations of the experiments are presented in Table 8.

Table 7: Runtime and Memory Usage

Runtime	Memory	GPU Memory
15 minutes	868 MB	4.33 GB

Table 8: Machine configuration.

Component	Specification
GPU	NVIDIA GeForce RTX 3090
CPU	Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to the abstract and the introduction section. Our claims are supported by the experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a limitation section at Section 7

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our theoretical analysis of complexity analysis is included in the Appendix A.2. The analysis includes contextual information to help understand each step. We provide no theorems in this paper, so no assumption is made.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We illustrate the information in the experimental settings section Section 5.1 and provide information for datasets in Appendix B.1. Codes are also provided in the supplementary material for verifying reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide codes in the supplementary material, with a readme file as guidance for setting up the environment and reproducing experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We present the experimental setting details in Section 5.1. The default setting in the provided code also aligns with the experimental setting described in this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Experiments in Figure 3, right subfigure report the runtime efficiency by reporting the mean and std (shown in error bars) from three rounds of repeated experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the machine configurations at the end of the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We checked and ensured that our paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper focuses on technical improvements in entity alignment and does not introduce direct societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks, we have not released datasets nor high-risk models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers of the datasets and EA models used in our experiments, ensuring proper credit and respect for their licenses and terms of use. For the open-source code and datasets, we have also stated the licenses they use in the readme file in our code repository.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code of the proposed framework is released and accessible at the supplementary material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.