

# Predicate Importance Estimation and Decoupled Rationale-Score Distillation for Entity Alignment

Keunha Kim<sup>1</sup>, Yoonjin Jang<sup>1</sup>, Hyeon-gu Lee<sup>2</sup>, Sihyung Kim<sup>2</sup>, Youngjoong Ko<sup>1\*</sup>

<sup>1</sup>SungKyunKwan University, <sup>2</sup>NAVER,

{keunhakim98, yoonjinjang98}@gmail.com, yjko@skku.edu

{hyeongu.lee, sihyung.kim}@navercorp.com

## Abstract

Knowledge graphs (KGs) are increasingly used as structured context for Large Language Models (LLMs), but industrial KG-RAG systems often need to integrate public and domain-specific KGs constructed from heterogeneous databases. This integration relies on Entity Alignment (EA), where lexical matching alone is insufficient under predicate-name variation and incomplete local neighborhoods. We address EA for KG integration by constructing a pairwise EA dataset and proposing two complementary modules: Predicate Importance Estimation (PIE) and Decoupled Rationale-Score Distillation (DRSD). PIE is a compact embedding-based approach that removes the subject information from each 1-hop triple, encodes the resulting subjectless triples, and aggregates them with learnable predicate-importance weights to build predicate-aware entity embeddings. DRSD trains a distilled small language model (SLM) with pseudo-answers produced by a teacher LLM through distinct prompts. By converting binary EA labels into text-based supervision and decoupling confidence-score estimation from label-consistent rationales, DRSD enables the SLM to learn task-specific reasoning while retaining a less label-biased confidence signal. Experiments show that PIE and DRSD improve EA classification. Moreover, because DRSD decouples confidence-score estimation from the decision, a division between the two flags an uncertain prediction for human review, thereby enabling a practical discrepancy between automatic acceptance and human-in-the-loop verification.

## 1 Introduction

Knowledge graphs (KGs) are increasingly used as structured knowledge sources for Large Language Models (LLMs) (Pan et al., 2024; Sun et al., 2024). Industrial systems simultaneously reason over public KGs such as DBpedia and Wikidata, as well

\*Corresponding author

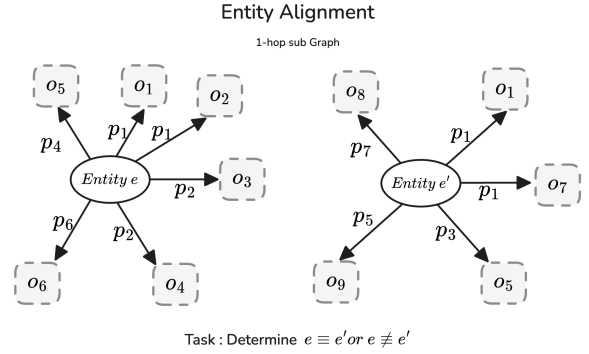


Figure 1: 1-hop graph example for an entity-alignment pair

as domain-specific KGs that are constructed from multiple internal relational databases via standard mappings (Das et al., 2012; Sequeda and Miranker, 2017). Therefore, these heterogeneous sources need to be integrated into a unified KG. To build such a unified KG, duplicate entities must be identified and merged across sources, which requires an explicit entity-alignment step. **Entity Alignment** (EA) identifies equivalent entities across distinct KGs (Sun et al., 2020b; Zhang et al., 2022). EA is intrinsically hard for two reasons that matter especially when systems bridge different graphs. First, two graphs typically use different predicate-naming conventions (e.g., `employer_of` vs. `worksFor`), so purely structural or string-based matching is misled by surface variance. Second, knowledge graphs collected for different purposes often contain different types and amounts of information. Therefore, robust EA must go beyond semantic similarity through joint modeling of the surrounding subgraph and the **relative importance** of each predicate as evidence for alignment.

To handle such complex semantic judgments, Large Language Models (LLMs) offer attractive generalization capabilities, and these capabilities have led recent work to exploit them for EA (Jiang

et al., 2024; Zhang et al., 2024; Chen et al., 2024). However, these LLM-based aligners rely heavily on general pretraining knowledge, so they can struggle to internalize the domain-specific relations and user-defined alignment constraints that define these heterogeneous graphs, patterns that are under-represented during pretraining. Because our KGs are domain-specific, we do not directly compare against these methods; instead, we target the same limitation by learning predicate importance directly from in-domain data rather than relying on an LLM’s pretrained knowledge. In this work, we address entity alignment for KG integration by constructing a pairwise EA dataset and introducing two complementary alignment modules. **Our dataset** reflects the pre-alignment setting: each instance contains two ambiguous candidate entities and their 1-hop subgraph evidence, so the model must decide equivalence using local graph context.

**Our first module, Predicate Importance Estimation (PIE)**, is a compact embedding-based approach that removes the subject from each 1-hop triple, encodes the subjectless triples independently, and aggregates them with learnable predicate-importance weights into a compact, predicate-aware entity embedding.

**Our second module, Decoupled Rationale-Score Distillation (DRSD)**, distills a small language model (SLM) using decoupled rationale-score pseudo-answers from a teacher LLM. Since binary labels lack the reasoning and confidence signals needed for text-generating models, DRSD elicits structured teacher responses through distinct prompts: a gold-label prompt produces a label-consistent decision and rationale, while a score prompt estimates similarity without seeing the gold label or producing a decision. The final SFT target combines the gold-label decision and justification with the score-prompt confidence, letting the student learn task-specific reasoning while retaining a less label-biased confidence signal.

Compared with the prompt-only LLM baseline, both modules improve EA performance: PIE raises Acc./F1 by 6.32/4.63 points, and DRSD by 7.66/6.76 points. Because DRSD decouples score from decision, flagging their disagreement for human review already raises F1 from 0.886 to 0.909 (13.3% of pairs). PIE, by contrast, judges similarity directly from predicate-weighted evidence rather than through reasoning, so cross-checking its decision against DRSD’s surfaces disagreements exactly where one model’s judgment is unreliable;

this gives an even stronger review signal, raising F1 further to 0.947 at 15.5% flagged.

## 2 Related Work

### 2.1 Entity Alignment and LLM-based Approaches

Entity Alignment (EA) has been studied through embedding-based and neighborhood-aware methods, including translation-style KG embeddings, cross-graph extensions, and GNN-based aligners (Sun et al., 2020b; Zhang et al., 2022; Chen et al., 2017; Sun et al., 2018; Wang et al., 2018; Wu et al., 2019; Sun et al., 2020a). These methods show that structural, attribute, and predicate-aware features are important for distinguishing equivalent entities across heterogeneous KGs (Trisedya et al., 2019; Xu et al., 2019; Zhang et al., 2022). More recent LLM-based approaches, such as ChatEA, AutoAlign, and LLM4EA, exploit the semantic reasoning ability of LLMs for EA (Jiang et al., 2024; Zhang et al., 2024; Chen et al., 2024). However, they can be brittle on domain-specific relations and costly to run for each candidate pair (Ling et al., 2025). Our work complements these lines of research by learning predicate-aware entity representations and distilling LLM alignment reasoning into a smaller model with a score-based confidence signal.

### 2.2 Knowledge Distillation from LLMs to Small Models

Knowledge distillation (Hinton et al., 2015) has recently been extended from logit-matching to rationale transfer. **Distilling Step-by-Step** (Hsieh et al., 2023) trains a small model jointly on LLM-generated labels and chain-of-thought rationales (Wei et al., 2022) to match teacher performance with far fewer parameters. Subsequent work (Magister et al., 2023; Ho et al., 2023) demonstrates that small models can be taught to reason via CoT distillation, and SCOTT (Wang et al., 2023) introduces a counterfactual objective to reduce shortcut imitation.

### 2.3 Predicate-Aware KG Embeddings

Beyond the translation-based and GNN-based methods above, several lines of work explicitly model relation or attribute importance. KBGAT (Nathani et al., 2019) introduces attention-based neighborhood weighting for embeddings; relation-aware attention is similarly central to MRAEA

(Mao et al., 2020) and RDGCN (Wu et al., 2019); and attribute-centric methods such as JAPE (Sun et al., 2017) and AttrE (Trisedya et al., 2019) treat predicates and literal attributes as first-class signals. PIE instead uses subjectless, predicate-aware representations with importance-weighted pooling to produce a compact entity signature.

### 3 Preliminaries

Let  $\mathcal{C} = \{G_1, \dots, G_M\}$  denote a collection of knowledge graphs (KGs). Each KG is defined as

$$G_m = (\mathcal{E}_m, \mathcal{P}_m, \mathcal{T}_m), \quad (1)$$

where  $\mathcal{E}_m$  is the entity set,  $\mathcal{P}_m$  is the predicate set, and  $\mathcal{T}_m$  is the triple set of the  $m$ -th KG. We assume that entities are unique within each KG, while equivalent entities may appear across different KGs.

Our task is to compare entities from two different KGs,  $G_m$  and  $G_n$  ( $m \neq n$ ), and decide whether they denote the same entity. E.g.,  $e \equiv e'$  or  $e \not\equiv e'$ ,  $e \in G_m, e' \in G_n$ . Each triple is denoted as  $(s, p, o)$ , where  $s, p$ , and  $o$  denote subject, predicate, and object, respectively. For an entity  $e \in \mathcal{E}_m$ , its outgoing 1-hop neighborhood  $\mathcal{N}_m(e)$  is

$$\mathcal{N}_m(e) = \{(s, p, o) \in \mathcal{T}_m \mid s = e\}. \quad (2)$$

For a cross-KG entity pair  $(e, e') \in \mathcal{E}_m \times \mathcal{E}_n$ , we compute its similarity score  $s_{e,e'}$  as

$$s_{e,e'} = \phi(\mathcal{N}_m(e), \mathcal{N}_n(e')), \quad (3)$$

where  $\phi(\cdot, \cdot)$  is an entity-pair scoring function. The final alignment decision  $\hat{y}_{e,e'}$  is then

$$\hat{y}_{e,e'} = \begin{cases} 1, & \text{if } s_{e,e'} \geq \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where  $\tau$  is a decision threshold,  $\hat{y}_{e,e'} = 1$  indicates  $e \equiv e'$ , and  $\hat{y}_{e,e'} = 0$  indicates  $e \not\equiv e'$ .

Thus the task is to retrieve similar cross-collection candidates and decide match/non-match for integration. We refer to  $p$  as a *relation* when  $o$  is another entity and as an *attribute* when  $o$  is a literal value; PIE’s subjectless triple encoding (Section 4.1) treats both uniformly as predicate-object evidence.

This task setting is deliberately narrower than open-world retrieval: because  $(e, e')$  is already a candidate pair produced by an upstream retriever,

our method is a *verification* step over 1-hop evidence rather than a ranking step over the full entity pool, which is why we report classification metrics (Accuracy, F1) rather than retrieval metrics (Hits@ $k$ , MRR) throughout this paper (Section 5.2).

## 4 Methodology

Two complementary components are used for entity alignment. The first one is Predicate Importance Estimation (PIE), a lightweight statistical module that learns which predicates are most informative for entity alignment. The second one is Decoupled Rationale-Score Distillation (DRSD), a reasoning-aware distillation framework that transfers decision patterns from a large teacher model to a smaller student model.

### 4.1 Predicate Importance Estimation

For the 1-hop neighborhood  $\mathcal{N}_m(e)$  of an entity  $e$ , Predicate Importance Estimation (PIE) constructs a representative embedding  $\mathbf{v}_e$  from local relational evidence. It first encodes subjectless triples, then pools triple representations by predicate, and finally aggregates predicate-level representations with predicate-importance weights. This process yields both an entity embedding and predicate-importance weights, and the weights indicate which predicates are most informative for alignment. PIE consists of three stages: (1) subjectless triple encoding, (2) predicate-wise pooling, and (3) predicate-importance pooling.

**Subjectless triple encoding.** For each outgoing triple  $(e, p, o) \in \mathcal{N}_m(e)$ , we remove the subject and encode only the predicate-object text. This reduces reliance on the surface name of  $e$  and makes entity comparison depend more on relational evidence:

$$\mathbf{h}_{p,o}^{(e)} = \text{Enc}(\text{text}(p) \oplus \text{text}(o)), \quad (5)$$

where  $\oplus$  denotes text concatenation. Let  $\mathcal{P}(e) = \{p \mid \exists o : (e, p, o) \in \mathcal{N}_m(e)\}$  be the set of predicates around  $e$ , and let  $\mathcal{O}_p(e) = \{o \mid (e, p, o) \in \mathcal{N}_m(e)\}$  be the object set for predicate  $p$ .

**Predicate-wise pooling.** Since an entity may have multiple triples with the same predicate, we apply mean pooling over their triple embeddings to obtain one predicate-level representation:

$$\mathbf{r}_p^{(e)} = \frac{1}{|\mathcal{O}_p(e)|} \sum_{o \in \mathcal{O}_p(e)} \mathbf{h}_{p,o}^{(e)}. \quad (6)$$

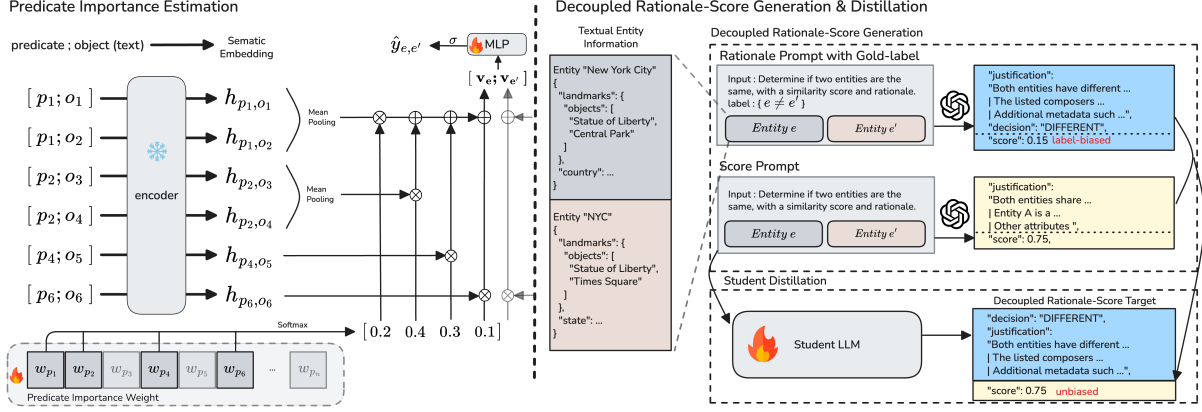


Figure 2: Overview of the proposed framework. (Left) Entity embedding construction ( $v_e$ ) within PIE; embeddings for each entity pair are subsequently concatenated for MLP-based classification. (Right) Illustrative example of the decoupled rationale-score distillation process.

**Predicate-importance pooling.** We then aggregate the predicate-level representations into a single entity embedding:

$$\alpha_p = \frac{\exp(w_p)}{\sum_{p' \in \mathcal{P}(e)} \exp(w_{p'})}, \quad (7)$$

$$\mathbf{v}_e = \sum_{p \in \mathcal{P}(e)} \alpha_p \mathbf{r}_p^{(e)}.$$

Here,  $w_p$  is a learnable scalar parameter for predicate  $p$ , and its softmax-normalized value  $\alpha_p$  represents the relative importance of that predicate within  $\mathcal{P}(e)$ . Predicates unseen during training receive a pre-softmax weight of zero by default (Appendix A).

**Entity-pair training.** For a candidate pair  $(e, e')$ , we concatenate the two entity embeddings and feed them into an MLP classifier:

$$\hat{y}_{e, e'} = \sigma(\text{MLP}([\mathbf{v}_e; \mathbf{v}_{e'}])), \quad (8)$$

where  $\sigma(\cdot)$  is the sigmoid function and  $\hat{y}_{e, e'} \in [0, 1]$  denotes the predicted match probability. We train the classifier with binary cross-entropy over entity pairs with labels, where  $y_{e, e'} = 1$  indicates that  $e$  and  $e'$  are equivalent, and  $y_{e, e'} = 0$  otherwise. During training, we freeze the pretrained encoder and update only the predicate-importance weights and the MLP classifier layers. This keeps the training overhead low and quickly adapts the predicate-importance weights to the alignment task. Appendix B reports a component-level ablation of these design choices.

## 4.2 Decoupled Rationale-Score Distillation

Conventional LLM distillation trains students to imitate teacher outputs, which makes the teacher an

implicit performance ceiling when its responses are incorrect. DRSD instead separates pseudo-answer generation into a supervised signal, where the teacher sees the gold label, and an unsupervised signal, where it does not. Combining the two preserves label-guided reasoning while yielding a confidence score that is less tied to the supervised answer.

### Decoupled Rationale-Score Label Construction

Our goal is to construct pseudo-answers that preserve the teacher’s reasoning ability while preventing the confidence score from being biased toward the gold label in the prompt, regardless of the objective similarity between the two entities. For the  $i$ -th labeled entity pair, let  $X_i$  denote the serialized evidence for the two entities and  $y_i$  its gold label. We use three prompt schemas:

$$\begin{aligned} q_i^{\text{gold}} &= \text{Prompt}_{\text{gold}}(X_i, y_i), \\ q_i^{\text{score}} &= \text{Prompt}_{\text{score}}(X_i), \\ q_i^{\text{sft}} &= \text{Prompt}_{\text{sft}}(X_i). \end{aligned} \quad (9)$$

The gold-label prompt generates a supervised decision and rationale, whereas the score prompt estimates confidence from the evidence without receiving the gold label or producing a decision. The SFT prompt is a simplified student training prompt for efficient fine-tuning. The teacher outputs are

$$\begin{aligned} \mathbf{o}_i^{\text{gold}} &= T(q_i^{\text{gold}}) = (j_i^{\text{gold}}, d_i^{\text{gold}}, c_i^{\text{gold}}), \\ \mathbf{o}_i^{\text{score}} &= T(q_i^{\text{score}}) = (j_i^{\text{score}}, c_i^{\text{score}}), \end{aligned} \quad (10)$$

where  $d$ ,  $c$ , and  $j$  denote the decision, score, and justification fields, respectively. Appendix K provides the prompt templates and representative JSON-style teacher responses, and Appendix C reports

a manual audit of teacher/student output format validity and score–justification consistency.

We then compose the final SFT target with the decision and justification from the gold-label output and the score from the score prompt output:

$$\ell_i^{\text{sft}} = (d_i^{\text{gold}}, c_i^{\text{score}}, j_i^{\text{gold}}). \quad (11)$$

These three fields correspond to the final alignment decision, confidence score, and reasoning justification, respectively. This decoupled rationale-score construction uses label information where it is most reliable—the decision and justification—and keeps confidence estimation independent of the gold label in the prompt. As a result, the score can be used as a more objective thresholding signal for precision–recall trade-offs and human-in-the-loop review.

#### 4.2.1 Predicate-Importance Weights

The predicate-importance weights learned by PIE are also used to compress LLM inputs. For a budget  $k$ , we keep only the triples whose predicates fall in the top- $k$  set,

$$\text{TopK}(\alpha, k) = \arg \max_{S \subseteq P(e), |S|=k} \sum_{p \in S} \alpha_p, \quad (12)$$

$$\tilde{\mathcal{N}}_m(e) = \{(e, p, o) \in \mathcal{N}_m(e) \mid p \in \text{TopK}(\alpha, k)\}, \quad (13)$$

where  $\alpha_p$  is the predicate-importance weight from Eq. (7). We use  $k=10$  and further retain at most 10 objects per predicate (Appendix F.4), which preserves informative evidence while reducing prompt length and inference cost.

#### 4.2.2 SFT

We construct an SFT dataset for a smaller language model as prompt–pseudo-answer pairs:

$$\mathcal{D}_{\text{SFT}} = \{(q_i^{\text{sft}}, \ell_i^{\text{sft}})\}_{i=1}^N. \quad (14)$$

The student model is fine-tuned to generate the decoupled rationale-score target response from the SFT prompt. In this way, the small language model learns the gold-label decision and justification, and it inherits the prompt-derived score as its confidence signal. At deployment time, the generated score can be thresholded to trade precision against recall according to application constraints. To analyze the effect of score decoupling, we compare DRSD against **Label-Guided Distillation (LGD)**, which is an ablation that uses a single gold-label prompt to jointly produce the decision, rationale,

| Method         | Acc.          | Prec.         | Rec.          | F1            |
|----------------|---------------|---------------|---------------|---------------|
| LLM            | 0.7976        | 0.7433        | 0.9092        | 0.8179        |
| SLM            | 0.7418        | 0.7737        | 0.6836        | 0.7258        |
| PIE(ours)      | 0.8608        | 0.8438        | 0.8856        | 0.8642        |
| DRSD(ours)     | 0.8742        | 0.8126        | 0.9728        | 0.8855        |
| + Human Review | <b>0.9020</b> | <b>0.8463</b> | <b>0.9824</b> | <b>0.9093</b> |

Table 1: Entity alignment performance on the evaluation split. LLM and SLM are prompt-only; DRSD is fine-tuned from the SLM. + *Human Review* corrects the 13.3% of pairs flagged as score–decision conflicts.

and score without decoupling. Appendix D reports LGD’s score–decision inconsistency and shows that exposing the gold label alone suppresses it well below DRSD’s, so review routing on LGD yields far less gain than on DRSD.

## 5 Experiments

### 5.1 Datasets & Models

**Datasets** We evaluate our method on a real-world, industrial-scale labeled entity-pair dataset constructed from our in-house heterogeneous knowledge graphs. The dataset contains a total of 85,858 training entity pairs and 5,000 evaluation pairs. Each pair is assigned a binary label: SAME for equivalent entities and DIFFERENT for non-equivalent entities. The evaluation split spans 102 source collections and 863 entity classes with a long-tailed domain distribution. A detailed description of the underlying knowledge graphs and dataset statistics, including 1-hop triple distributions, is provided in Appendix E.

**Models** For PIE, we use BAAI/bge-m3<sup>1</sup> to generate embeddings and employ a two-layer MLP as the classifier. For DRSD, we use openai/gpt-oss-120b<sup>2</sup> as the teacher LLM and train a hcx-srch1-8B (In-house SLM) for distillation. Detailed architecture and training hyperparameters are provided in Appendix F.

### 5.2 Results

Table 1 reports entity alignment performance on the evaluation split. For DRSD, the model outputs both a decision and a similarity score. We compute the reported metrics from the decision output by treating SAME predictions as the positive label.

The results highlight two main findings. First, PIE achieves strong performance despite its com-

<sup>1</sup><https://huggingface.co/BAAI/bge-m3>

<sup>2</sup><https://huggingface.co/openai/gpt-oss-120b>

| Method                        | Acc.         | F1           |
|-------------------------------|--------------|--------------|
| PIE (ours)                    | <b>0.861</b> | <b>0.864</b> |
| GCN-Align (Wang et al., 2018) | 0.787        | 0.801        |
| AliNet (Sun et al., 2020a)    | 0.776        | 0.780        |
| LargeEA (Ge et al., 2021)     | 0.727        | 0.755        |
| Dual-AMN (Mao et al., 2021)   | 0.703        | 0.759        |

Table 2: Comparison with structure-based entity-alignment baselines on our evaluation split. Each baseline’s ranking output is converted into a binary decision with a logistic head over cosine similarity at threshold 0.5, matching PIE’s decision rule (Appendix H).

pact architecture and small number of trainable parameters. This suggests that subjectless triples can be effectively encoded in the semantic embedding space and aggregated with predicate-aware weights to construct discriminative entity representations for classification.

Second, the LLM–DRSD comparison shows that pseudo-answer distillation is effective: DRSD, the distilled version of the SLM, outperforms the prompted LLM and implicitly learns task-specific alignment rules embedded in the data. DRSD inherits a prior distribution biased toward SAME from the imbalanced training split (76.8% SAME) and thus prioritizes recall, whereas PIE is more balanced; unlike the prompt-only LLM, PIE’s recall is largely unaffected by shared entity names, indicating that training prevalence—not name overlap—drives DRSD’s recall skew (Appendix G). Finally, routing DRSD’s uncertain predictions to human review raises F1 from 0.886 to 0.909 (Table 1); we analyze why in Section 5.4.

### Comparison with Structure-Based EA Baselines

Table 2 compares PIE against four representative structure-based EA methods (GCN-Align, AliNet, LargeEA, and Dual-AMN). Since these methods are designed to rank candidates rather than classify a given pair, we convert each into a binary classifier that matches PIE’s decision rule (Appendix H). As discussed in Section 3, we report Accuracy and F1 rather than Hits@ $k$ /MRR. PIE outperforms all four baselines by a wide margin, indicating that predicate-aware subjectless triple encoding captures alignment evidence that these structure-only representations do not.

### 5.3 Analysis

**Predicate Budget  $k$**  Table 3 reports prompt-only teacher LLM classification performance as the predicate budget  $k$  in PIE’s importance-based

| $k$ | Acc.          | Prec.         | Rec.          | F1            |
|-----|---------------|---------------|---------------|---------------|
| 5   | 0.7956        | 0.7340        | <b>0.9272</b> | 0.8194        |
| 10  | 0.7976        | 0.7433        | 0.9092        | 0.8179        |
| 15  | 0.8044        | 0.7532        | 0.9056        | 0.8224        |
| 20  | <b>0.8094</b> | <b>0.7574</b> | 0.9104        | <b>0.8269</b> |

Table 3: LLM classification results across top  $k$  predicate budgets. Best scores are shown in bold.

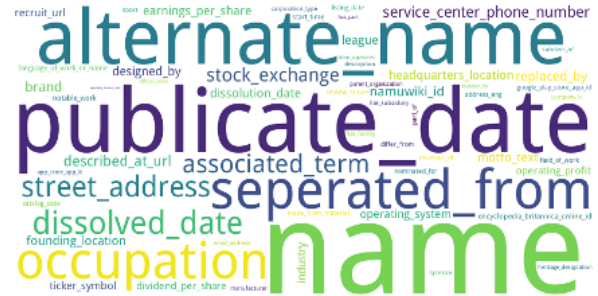


Figure 3: Predicate-importance weights learned by PIE for predicates in the specific domain (Company).

compression varies (Section 4.2.1). Performance changes only marginally as  $k$  grows from 5 to 20, while prompt length increases with  $k$ . This indicates that the top-ranked predicates already carry most of the alignment-relevant evidence, which motivates selecting  $k=10$  to keep the prompt short while maintaining accuracy.

**Predicate-importance Weights** Figure 3 visualizes the predicate-importance weights that PIE learns in the company domain. The highly weighted predicates correspond to attributes and relations describing organizational identity or business context, indicating that PIE learns interpretable predicate preferences consistent with human judgment. Appendix I further visualizes PIE’s pooled entity embeddings with t-SNE.

### 5.4 Human-in-the-Loop Review

We compare three routing strategies at a broadly similar review budget (13.3–15.5% flagged): random sampling, DRSD’s *internal* score–decision disagreement, and *external* cross-model PIE–DRSD decision disagreement. Because DRSD’s score is decoupled from its decision, we first test whether internal disagreement alone can approximate the benefit of comparing against an independent second model. Figure 4 compares where LGD and DRSD place their own errors across score-bins (false negatives at top, false positives at bottom). Because LGD’s prompt exposes the gold label, its score stays close to the decision, so most of its

false negatives and false positives fall on the expected side of the threshold, and the resulting score–decision discrepancy (2.97%, Appendix D) is a weak signal for unreliable predictions. DRSD instead decouples the score from the decision; among the resulting disagreements, the decoupled score can assign high scores to 24 pairs predicted as DIFFERENT and low scores to 115 pairs predicted as SAME. More broadly, these discrepancies concentrate on the model’s errors and form a concrete criterion for review, flagging 13.3% of the evaluation set.

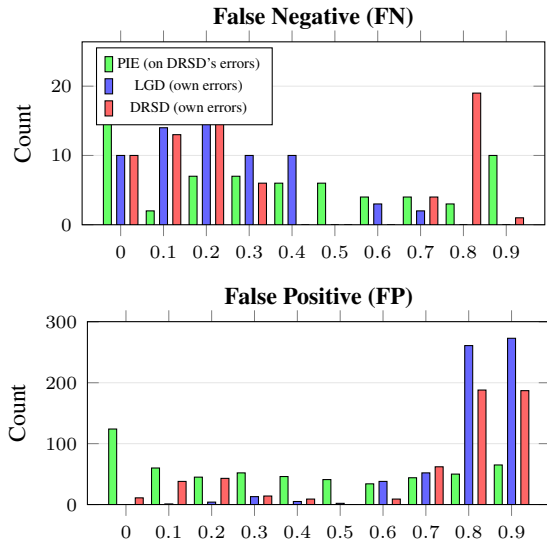


Figure 4: Score-bin distribution of false negatives (top) and false positives (bottom). Red (DRSD) and blue (LGD) show each model’s own score on its own errors; green (PIE) shows PIE’s score on DRSD’s error pairs, which often lands opposite DRSD’s wrong decision.

Table 4 places this internal disagreement rule alongside random sampling and external PIE–DRSD disagreement at a broadly similar review budget. Internal disagreement clearly beats random sampling (F1 0.9093 vs. 0.8997), confirming that the decoupled score carries real signal, but it still falls short of external disagreement: Figure 4 shows that PIE’s independently computed score on these same DRSD error pairs frequently lands on the side opposite DRSD’s wrong decision (high for its false negatives, low for its false positives), and flagging this external disagreement more than doubles internal disagreement’s precision at a similar flagging rate, raising F1 further to 0.947 at 15.5% flagged (full comparison in Appendix J). This gap suggests that the value of cross-model disagreement lies not simply in having a second score, but in having a second, *structurally different* model: PIE

| Routing rule                 | Flagged       | Flag prec.   | Acc.          | F1            |
|------------------------------|---------------|--------------|---------------|---------------|
| No review (baseline)         | —             | —            | 0.8742        | 0.8855        |
| Random sampling              | 13.34%        | 0.127        | 0.8911        | 0.8997        |
| DRSD disagreement            | 13.34%        | 0.208        | 0.9020        | 0.9093        |
| <b>PIE–DRSD disagreement</b> | <b>15.50%</b> | <b>0.457</b> | <b>0.9450</b> | <b>0.9470</b> |

Table 4: Human-review outcomes for three routing strategies at a similar flagging rate (13.3–15.5%). *Flag prec.* is the fraction of flagged pairs that were actual errors. PIE–DRSD disagreement achieves over 2× the precision at a similar budget, yielding the largest F1 gain (Appendix J).

is a lightweight, non-reasoning embedding scorer, while DRSD is a distilled reasoning model, and the two tend to fail in different ways. Internal disagreement can only surface cases where DRSD itself is unsure; it cannot catch cases where DRSD is confidently wrong. External disagreement catches exactly these cases, because PIE’s structural evidence and DRSD’s learned reasoning rarely fail on the same pairs.

## 6 Conclusion

We presented an entity-alignment framework for heterogeneous KG integration that combines Predicate Importance Estimation (PIE) and Decoupled Rationale-Score Distillation (DRSD). PIE learns predicate-aware entity representations from subjectless 1-hop triples, while DRSD distills LLM reasoning into a smaller model by decoupling label-consistent rationales from confidence-score estimation. Because DRSD’s score is decoupled from its decision, disagreements—especially external disagreement with PIE’s independently trained decision—flag likely errors for human review, raising F1 from 0.886 to 0.947 while flagging 15.5% of pairs. We adopt this external, cross-model disagreement as our primary deployment signal because it needs no extra model beyond the two already in the pipeline and offers an independent second opinion rather than relying on a single model’s internal score–decision consistency; in an industrial setting where false-positive merges are costly and review capacity is limited, this lets an operator accept confident predictions automatically and focus review on genuinely uncertain cases. This decoupling principle is not limited to EA and can also be useful in other human-in-the-loop distillation settings.

## Limitations

Our study has several limitations. First, although PIE is designed to learn predicate-importance

weights, these weights can become biased when some classes or predicates dominate the training data. Ideally, predicate importance should be estimated in a class-aware manner, but predicate distributions also vary across collections. Future work should explore multiple importance weights, ensemble-based weighting, or other debiasing strategies that produce predicate weights generalizing across both classes and collections.

Second, our human-review results assume correct annotator judgment and therefore represent an upper bound on the achievable gain. In practice, the realized improvement depends on annotator accuracy and the available review budget, and the 13.3–15.5% of pairs flagged for review may be costly to inspect at scale. Estimating the gain under realistic annotator behavior is left to future work.

Third, PIE uses only 1-hop predicate-object evidence and does not model deeper graph structure, which places it methodologically closer to entity-matching approaches over flat attribute-relation records (e.g., Ditto, DeepMatcher) than to multi-hop, structure-heavy EA methods. We compare against structure-based EA baselines (Section 5.2) because they are converted into the same pairwise verification setting as our task, not because PIE targets their multi-hop retrieval setting; extending PIE to  $n$ -hop evidence is left to future work.

## Ethical Considerations

**Data.** Our experiments use proprietary, in-house knowledge graphs constructed from internal databases. The data contains entity records such as organizations, people, and creative works but no private personal information beyond what is publicly associated with these entities. Because the data is proprietary, we do not release it; we instead report dataset statistics (Appendix E) and full implementation details (Appendix F) to support reproducibility of the method.

**Human review.** The “+ Human Review” row in Table 1 and the routing rows in Table 4 and Table 13 use the manually verified evaluation labels as the outcome of reviewing the flagged pairs. Because these labels are human-verified, the result can be interpreted as a simulation of correct human review on the flagged subset. We report it separately from fully automatic methods because deployment gains would still depend on annotator accuracy, review capacity, and quality-control procedures.

**Intended use.** The proposed method targets knowledge-graph integration in industrial settings. As with any automated alignment system, incorrect merges could propagate errors into downstream applications. The confidence-based routing we propose is designed precisely to mitigate this risk by escalating uncertain cases to human review.

## Acknowledgments

This work was supported in part by [Technical Consulting for Advancing Generative AI-based Search Technology] funded by NAVER corporation, South Korea (60), Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II220369, (Part 4) Development of AI Technology to support Expert Decision-making that can Explain the Reasons/Grounds for Judgment Results based on Expert Knowledge, 20), Institute of Information & Communications Technology Planning & Evaluation(IITP)-ICT Creative Consilience Program grant funded by the Korea government(MSIT)(IITP-2026-RS-2020-II201821, 10) and Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2019-II190421, AI Graduate School Support Program(Sungkyunkwan University), 10)

## References

- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. [Multilingual knowledge graph embeddings for cross-lingual knowledge alignment](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1511–1517.
- Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. 2024. [Entity alignment with noisy annotations from large language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Souripriya Das, Seema Sundara, and Richard Cyganiak. 2012. [R2RML: Rdb to rdf mapping language](#). W3C Recommendation.
- Congcong Ge, Xiaoze Liu, Lu Chen, Yunjun Gao, and Baihua Zheng. 2021. [Largeea: aligning entities for large-scale knowledge graphs](#). *Proc. VLDB Endow.*, 15(2):237–245.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *arXiv preprint arXiv:1503.02531*.

- Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. [Large language models are reasoning teachers](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. [Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.
- Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. 2024. [Unlocking the power of large language models for entity alignment](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7566–7583, Bangkok, Thailand. Association for Computational Linguistics.
- Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Dhagash Mehta, Stefano Pasquali, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, and 5 others. 2025. [Domain specialization as the key to make large language models disruptive: A comprehensive survey](#). *ACM Comput. Surv.*, 58(3).
- Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. [Teaching small language models to reason](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.
- Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. [Boosting the speed of entity alignment 10 ×: Dual attention matching network with normalized hard sample mining](#). In *Proceedings of the Web Conference 2021*, WWW '21, page 821–832. ACM.
- Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. [Mraea: An efficient and robust entity alignment approach for cross-lingual knowledge graph](#). In *Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20*, page 420–428, New York, NY, USA. Association for Computing Machinery.
- Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. [Learning attention-based embeddings for relation prediction in knowledge graphs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. [Unifying large language models and knowledge graphs: A roadmap](#). *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.
- Juan F. Sequeda and Daniel P. Miranker. 2017. [A pay-as-you-go methodology for ontology-based data access](#). *IEEE Internet Computing*, 21(2):92–96.
- Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. [Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph](#). In *The Twelfth International Conference on Learning Representations*.
- Zequan Sun, Wei Hu, and Chengkai Li. 2017. [Cross-lingual entity alignment via joint attribute-preserving embedding](#). In *The Semantic Web – ISWC 2017*, pages 628–644, Cham. Springer International Publishing.
- Zequan Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. [Bootstrapping entity alignment with knowledge graph embedding](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4396–4402. International Joint Conferences on Artificial Intelligence Organization.
- Zequan Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020a. [Knowledge graph alignment network with gated multi-hop neighborhood aggregation](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 222–229.
- Zequan Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020b. [A benchmarking study of embedding-based entity alignment for knowledge graphs](#). *Proc. VLDB Endow.*, 13(12):2326–2340.
- Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. [Entity alignment between knowledge graphs using attribute embeddings](#). In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19*. AAAI Press.
- Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. [SCOTT: Self-consistent chain-of-thought distillation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.
- Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. [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,

Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. [Relation-aware entity alignment for heterogeneous knowledge graphs](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5278–5284. International Joint Conferences on Artificial Intelligence Organization.

Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. [Cross-lingual knowledge graph alignment via graph matching neural network](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3156–3161, Florence, Italy. Association for Computational Linguistics.

Rui Zhang, Yixin Su, Bayu Distiawan Trisedya, Xiaoyan Zhao, Min Yang, Hong Cheng, and Jianzhong Qi. 2024. [Autoalign: Fully automatic and effective knowledge graph alignment enabled by large language models](#). *IEEE Transactions on Knowledge and Data Engineering*, 36(6):2357–2371.

Rui Zhang, Bayu Distiawan Trisedya, Miao Li, Yong Jiang, and Jianzhong Qi. 2022. [A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning](#). *The VLDB Journal*, 31(5):1143–1168.

## A Unseen-Predicate Handling

Predicates absent from training are rare in our evaluation split (about 0.7%, 33 pairs). PIE assigns such predicates a pre-softmax weight of zero by default. Table 5(a) compares this default against two alternatives on the 33 affected pairs: an attention-reconstructed weight (inherited from the most similar seen predicate’s embedding) and dropping the predicate’s triples entirely. All three schemes give identical performance on this small subset, so we retain the simpler zero-weight default; because the subset is so small, we do not draw a statistically grounded comparison against the full evaluation set’s performance here and leave a larger-scale breakdown to future work. To validate the reconstruction idea on a larger sample, Table 5(b) reconstructs weights for a held-out 20% of predicates (213 pairs) from their similarity to seen predicates: the reconstructed weight matches the originally learned weight and outperforms dropping the predicate, though the sample size keeps the effect size modest.

| (a) Weight assignment on 33 unseen-predicate pairs |       |       |
|----------------------------------------------------|-------|-------|
| Scheme                                             | Acc.  | F1    |
| Zero weight (ours)                                 | 0.758 | 0.600 |
| Attention-inherited weight                         | 0.758 | 0.600 |
| Drop the predicate                                 | 0.758 | 0.600 |

| (b) Held-out predicate reconstruction (213 pairs) |       |       |
|---------------------------------------------------|-------|-------|
| Scheme                                            | Acc.  | F1    |
| Current model (zero weight)                       | 0.798 | 0.726 |
| Attention-reconstructed weight                    | 0.798 | 0.726 |
| Drop the rare predicate                           | 0.775 | 0.700 |

Table 5: Unseen-predicate handling. (a) Only  $\sim 0.7\%$  of evaluation pairs (33) contain a predicate absent from training; assigning it zero pre-softmax weight, an attention-reconstructed weight, or dropping it entirely give identical performance on this small subset, so we retain the simpler zero-weight default. (b) On a larger held-out set of 213 pairs restricted to rare predicates (20% of predicates withheld and reconstructed by similarity to seen predicates), the reconstructed weight matches the originally learned weight and outperforms dropping the predicate, though the sample size keeps the effect size small.

## B PIE Component Ablation

Table 6 reports a component-level ablation of PIE covering (a) subject removal from each triple, (b) the predicate-importance weighting scheme, (c) the entity-embedding aggregation strategy, and (d) the pair-classifier head architecture. Removing the subject improves performance because candidate pairs are already retrieved by alternate-name similarity,

| (a) Triple encoding |              |              |
|---------------------|--------------|--------------|
| Variant             | Acc.         | F1           |
| Subjectless (ours)  | <b>0.861</b> | <b>0.864</b> |
| With subject        | 0.825        | 0.837        |

| (b) Predicate weighting             |              |              |
|-------------------------------------|--------------|--------------|
| Variant                             | Acc.         | F1           |
| Learned scalar per predicate (ours) | <b>0.861</b> | <b>0.864</b> |
| Uniform                             | 0.850        | 0.858        |
| Evidence-conditioned attention      | 0.851        | 0.863        |

| (c) Node encoding                 |              |              |
|-----------------------------------|--------------|--------------|
| Variant                           | Acc.         | F1           |
| Predicate-wise pooling (ours)     | <b>0.861</b> | 0.864        |
| Max-pooling                       | 0.856        | <b>0.865</b> |
| Flat mean (no predicate grouping) | 0.845        | 0.855        |
| Random single triple              | 0.711        | 0.748        |

| (d) Pair-classifier head        |              |              |
|---------------------------------|--------------|--------------|
| Variant                         | Acc.         | F1           |
| Cosine similarity               | 0.500        | 0.667        |
| MLP, depth 1                    | 0.576        | 0.693        |
| MLP, depth 2, width 256         | 0.844        | 0.857        |
| MLP, depth 2, width 1024 (ours) | <b>0.861</b> | <b>0.864</b> |
| MLP, depth 2, width 2048        | 0.856        | 0.862        |
| MLP, depth 3, width 1024        | 0.850        | 0.859        |

Table 6: Component-level ablation of PIE. (a) Removing the subject from each triple improves performance because candidate pairs are already retrieved by alternate-name similarity, so re-encoding near-identical subject names is redundant. (b) Learned per-predicate scalars perform comparably to an evidence-conditioned attention variant while remaining well-defined for unseen predicates (Appendix A). (c) Predicate-wise mean pooling attains higher Accuracy than max-pooling (with marginally lower F1), and both clearly outperform schemes that discard predicate structure (flat mean, single triple). (d) A depth-1 head collapses to an all-positive classifier; a 2-layer MLP is sufficient and a 3rd layer does not help.

so the subject names of the two entities are nearly identical and re-encoding them is redundant; the model instead focuses on the more discriminative predicate-object evidence. For predicate weighting, we additionally compare our learned per-predicate scalar against an evidence-conditioned attention variant, whose weights are conditioned on the predicate and object rather than looked up by predicate identity; the two perform comparably, and the attention variant has the added benefit of being well-defined for predicates unseen during training without requiring the lookup-table extension discussed in Appendix A.

## C LLM Output Validation

All 5,000 test-set outputs from the DRSD student are valid JSON containing the three expected fields (decision, score, justification). We additionally

| Variant             | SAME & low-score | DIFF & high-score | Total           |
|---------------------|------------------|-------------------|-----------------|
| LGD (label exposed) | 1,330 (2.02%)    | 1,218 (6.13%)     | 2,548 (2.97%)   |
| DRSD (score-only)   | 7,388 (11.20%)   | 5,799 (29.18%)    | 13,187 (15.36%) |

Table 7: Train-set score–decision inconsistency at threshold  $t=0.60$  (gold SAME=65,984, DIFFERENT=19,874). Because LGD’s prompt exposes the gold label, its score is biased toward the decision, so its inconsistency (2.97%) is far below DRSD’s label-free 15.36%.

| Student | Setting                   | Acc.          | Prec.         | Rec.          | F1            |
|---------|---------------------------|---------------|---------------|---------------|---------------|
| LGD     | base                      | 0.8556        | 0.7890        | 0.9708        | 0.8705        |
| LGD     | + review (1.06% flagged)  | 0.8616        | 0.7958        | 0.9728        | 0.8754        |
| DRSD    | base                      | 0.8742        | 0.8126        | 0.9728        | 0.8855        |
| DRSD    | + review (13.34% flagged) | <b>0.9020</b> | <b>0.8463</b> | <b>0.9824</b> | <b>0.9093</b> |

Table 8: Score–decision-disagreement routing applied to each variant. LGD flags only 1.06% of pairs (versus DRSD’s 13.34%) and gains  $\Delta F1 = 0.0049$ , far smaller in absolute terms than DRSD’s  $\Delta F1 = 0.0238$ , because exposing the gold label during training suppresses the score–decision discrepancy that human-review routing relies on.

manually audit 169 pairs—100 sampled uniformly at random plus 69 flagged by a lexical scan for potential justification/decision mismatches—and find only one inconsistency (a score-side, not decision-side, mismatch). Comparing the student’s outputs against the teacher’s gold-label output across the full 5,000-pair test set, decision agreement is 87.4% and score mean absolute error is 0.068, indicating that the student reproduces the teacher’s targets closely.

## D LGD Ablation

Section 5.4 shows that LGD produces substantially less score–decision discrepancy than DRSD, which we use as motivation for DRSD’s decoupled construction. Table 7 reports the resulting train-set score–decision inconsistency: because LGD’s prompt exposes the gold label, its inconsistency (2.97%) stays far below DRSD’s label-free 15.36%. Table 8 confirms this has practical consequences: applying score–decision-disagreement routing to LGD flags only 1.06% of pairs (versus DRSD’s 13.34%) and gains  $\Delta F1 = 0.0049$ , far smaller in absolute terms than DRSD’s  $\Delta F1 = 0.0238$ , because gold-label exposure suppresses the score–decision disagreement that this routing signal depends on.

## E Data Statistics

**Knowledge Graphs** Our entity pairs are drawn from in-house heterogeneous knowledge graphs

spanning multiple domains and sources. The graphs encompass entities from diverse domains such as entertainment (films, music, and broadcasts), companies, people, and cultural properties, connected through 2,904 distinct relation types. Because the graphs are heterogeneous, entities vary in their types and in the relations connecting them, so the set of 1-hop triples available for an entity differs substantially in both quantity and schema across the dataset. For each entity in a pair, we extract its 1-hop triples as the structural context provided to the model.

**Statistics** Table 9 reports the dataset scale, label distribution, and 1-hop triple statistics for the training and evaluation splits. The labeled training split is imbalanced toward SAME pairs, which account for 76.8% of training examples, whereas the evaluation split is intentionally balanced with a 50/50 label ratio. For the triple statistics, Triples/entity (mean) indicates the average number of 1-hop triples available per entity, while Triples/entity (max) indicates the largest number of 1-hop triples observed in each split.

| Statistic             | Train          | Evaluation    |
|-----------------------|----------------|---------------|
| Entity pairs          | 85,858         | 5,000         |
| SAME                  | 65,984 (76.8%) | 2,500 (50.0%) |
| DIFFERENT             | 19,874 (23.2%) | 2,500 (50.0%) |
| Unique entities       | 136,626        | 9,499         |
| Triples/entity (mean) | 26.17          | 25.0          |
| Triples/entity (max)  | 4,349          | 1,811         |

Table 9: Dataset statistics for the labeled training and evaluation splits. Triples/entity reports the mean and maximum number of 1-hop triples per entity.

**Entity-type coverage.** Table 10 reports the domain and entity-class coverage of the evaluation split. The distribution is long-tailed: the four largest entity classes (business enterprise, animation/TV, person, and cultural heritage) together account for 46.3% of entity–class assignments (4,676 of 10,095), with the remainder spread across 859 additional classes.

**Evaluation prevalence.** We intentionally balance the evaluation split 50/50 because, in deployment, a false-positive merge (incorrectly predicting SAME) is more costly than a missed merge, and evaluating at the training prevalence (76.8% SAME) could let a majority-class-biased model appear stronger than it is. Table 11 confirms that this 50/50 choice is the more conservative one: both PIE and DRSD score substantially higher when the

| Statistic           | Value     |
|---------------------|-----------|
| Source collections  | 102       |
| Entity classes      | 863       |
| Business enterprise | 1,533     |
| Animation / TV      | 1,085     |
| Person              | 1,080     |
| Cultural heritage   | ~980      |
| Other (859 classes) | remainder |

Table 10: Domain and entity-class coverage of the 5,000-pair (9,499-entity) evaluation split, tallied as 10,095 entity–class assignments since an entity may belong to more than one class. The four largest classes are shown; the remainder are long-tail distributed across 859 additional classes.

| Model | Prevalence    | Acc.  | Prec. | Rec.  | F1    |
|-------|---------------|-------|-------|-------|-------|
| PIE   | 50:50 (eval)  | 0.861 | 0.844 | 0.886 | 0.864 |
| PIE   | 75:25         | 0.873 | 0.942 | 0.886 | 0.913 |
| PIE   | 76:24 (train) | 0.874 | 0.947 | 0.886 | 0.915 |
| DRSD  | 50:50 (eval)  | 0.874 | 0.813 | 0.973 | 0.886 |
| DRSD  | 75:25         | 0.924 | 0.929 | 0.973 | 0.950 |
| DRSD  | 76:24 (train) | 0.927 | 0.935 | 0.973 | 0.954 |

Table 11: Sensitivity of PIE and DRSD to the SAME:DIFFERENT prevalence of the evaluation pool, obtained by resampling DIFFERENT pairs. Both models improve substantially at the actual training prevalence (76.8% SAME), confirming that our intentionally balanced 50/50 evaluation split is the more conservative choice.

evaluation pool is resampled toward the training prevalence. We do not have a separately measured deployment-time prevalence; the training split’s 76.8% SAME rate, drawn from our alternate-name-similarity retriever’s candidate pairs, is our closest estimate. The 2,500 DIFFERENT pairs are sampled uniformly at random from the candidate pool, without near-miss or difficulty-based filtering.

## F Implementation Details

### F.1 Decoupled Rationale-Score Distillation

The teacher LLM is openai/gpt-oss-120b, queried through the prompts in Appendix K; it is used offline for SFT-target generation only and is not part of the deployed pipeline (Table 12). The student is hcx-srch1-8B (SLM), fine-tuned as described below. We fine-tune the student SLM using the SFT objective described in the main method, with DeepSpeed ZeRO Stage 2, bfloat16 precision, and gradient checkpointing. We train for 1 epoch with AdamW (learning rate  $2 \times 10^{-5}$ , 100 warmup steps, no weight decay), an effective batch

size of 8, and a maximum sequence length of 4096 tokens, with a fixed random seed for data shuffling and initialization. We apply completion-only loss masking, discard over-length samples, and use greedy decoding with up to 256 new tokens at inference time. During inference, the decision field (SAME/DIFFERENT) is generated directly by the student and used for match classification, while the separately generated score field is retained for threshold-based confidence analysis (e.g.,  $t=0.60$  in Appendix D and Table 13).

### F.2 Runtime

Table 12 reports the per-component inference runtime of the deployed pipeline, measured on a single A100 GPU. Triple encoding with BGE-M3 is cached once per entity and reused across candidate pairs; the teacher LLM is used only offline to generate SFT targets and is excluded from the deployed runtime.

### F.3 Predicate Importance Estimation (PIE)

All predicate–object texts are encoded offline with the BAAI/bge-m3 encoder. For each candidate pair, the resulting entity embeddings are concatenated and passed to a two-layer MLP classifier. The pre-trained encoder is frozen, and only the predicate-importance weights and MLP classifier layers are trained with binary cross-entropy using AdamW (learning rate  $1 \times 10^{-3}$ , 100 warmup steps, no weight decay) for up to 20 epochs; validation accuracy selects the best checkpoint. We fix the decision threshold  $\tau$  (Eq. (4)) at 0.5 and do not tune it post hoc on the validation set; because the classifier is trained with a sigmoid output and binary cross-entropy loss, 0.5 is the natural decision boundary.

### F.4 Pseudo-Answer Construction

Each entity’s 1-hop neighborhood is serialized as JSON-style evidence, excluding collection-type objects and retaining at most 10 objects per predicate to control prompt length. For each labeled candidate pair, the teacher LLM (openai/gpt-oss-120b) is queried with both the gold-label prompt and the score prompt. The final SFT target follows the main decoupled rationale-score construction: the decision and justification are taken from the gold-label output, while the score is taken from the score prompt output. Consequently, the DRSD student’s training data consists of the full set of 85,858 gold-labeled training pairs (Table 9), each paired with a teacher-generated (i.e.,

| Component                       | Runtime           | Role                                                                 |
|---------------------------------|-------------------|----------------------------------------------------------------------|
| make_description (CPU)          | 0.086 ms/pair     | Predicate-importance input compression                               |
| BGE-M3 triple encoding (cached) | 1,147 triples/s   | PIE feature extraction, cached once                                  |
| PIE head                        | 2.9 ms/pair       | SAME/DIFFERENT scorer (F1 0.864)                                     |
| DRSD student (8B, vLLM)         | 60.7 ms/pair      | Decision + confidence + justification (F1 0.886 $\rightarrow$ 0.909) |
| Teacher LLM (120B)              | offline, one-time | SFT target generation only; not in the deployed pipeline             |

Table 12: Per-component inference runtime, measured on a single A100 GPU.

pseudo-labeled) decoupled rationale-score target rather than the raw binary label; we validate the fidelity of these pseudo-labels against the teacher and against manual review in Appendix C.

## G Recall Skew Analysis

We analyze why DRSD and the prompt-only LLM prioritize recall over precision. In our evaluation split, 58.3% (1,458/2,500) of DIFFERENT pairs share an identical entity name. The prompt-only LLM misclassifies 40.9% of this subset as SAME, versus 18.0% for DIFFERENT pairs without a shared name; 76% of its false positives (597/785) come from the shared-name subset, and removing name information from its input drops recall from 0.909 to 0.624. PIE is far less affected by the same intervention: its false positive rate on shared-name pairs is nearly unchanged (19.5%  $\rightarrow$  20.2%) and its recall drops by only 0.059, about a fifth of the LLM’s drop, because “name” is only one of several predicates combined through PIE’s softmax-weighted pooling. DRSD’s recall skew has a different cause: it is not driven by shared names but by the 76.8% SAME training prevalence. Retraining DRSD on a 50/50 rebalanced training set lowers recall from 0.9728 to 0.9016 but raises both Accuracy (0.8742  $\rightarrow$  0.8910) and F1 (0.8855  $\rightarrow$  0.8921), confirming that DRSD’s recall skew is a prevalence effect rather than a name-matching shortcut.

## H Structure-Based EA Baseline Details

The four baselines in Table 2 (GCN-Align (Wang et al., 2018), AliNet (Sun et al., 2020a), LargeEA (Ge et al., 2021), and Dual-AMN (Mao et al., 2021)) were originally released with TensorFlow 1.x implementations that do not run on our CUDA-12 hosts, so we re-implemented each in PyTorch following the original papers. Because these methods are designed to *rank* candidates rather than classify a given pair, we convert each into a binary classifier that matches our task: we embed both entities with the trained model, compute their cosine similarity, and pass it through a logistic head

thresholded at 0.5, the same decision rule used by PIE (Section 4.1). This conversion evaluates the baselines’ learned representations under our verification setting rather than their original open-world retrieval setting, for which Hits@ $k$ /MRR would be the appropriate metric (Section 3).

## I Qualitative Analysis of PIE Embeddings

We qualitatively inspect PIE’s pooled entity embeddings  $\mathbf{v}_e$  with t-SNE (Figure 5). At the domain level, cultural-heritage and business entities form well-separated clusters, while person and media entities partially overlap, reflecting genuine cross-domain relations (e.g., a person’s involvement in a media work). Within the media domain, the embeddings further separate into finer-grained sub-clusters corresponding to film/video, music, performance, and broadcast/character entities, indicating that PIE’s predicate-aware pooling captures sub-domain structure beyond what is needed for the binary alignment objective alone.

## J Routing Rule Definitions

Table 13 compares PIE–DRSD disagreement against four alternative routing rules: thresholding DRSD’s score or PIE’s probability near its own decision boundary, a random baseline matched to the same review budget, and DRSD’s own self-disagreement rule (Section 5.4). All rules are evaluated on the same 5,000-pair evaluation set with DRSD’s decision threshold fixed at  $t=0.60$ . *Flagged* is the number of pairs a rule routes to human review; *Errors caught* is the number of those flagged pairs that DRSD originally misclassified (629 pairs in total); *Flag precision* is errors caught divided by flagged pairs; and *Error recall* is errors caught divided by DRSD’s 629 total base errors. The random baseline is averaged over 500 random flagging seeds at the same budget (667 pairs) as DRSD’s self-disagreement rule, and we also report its best single-seed run for reference.

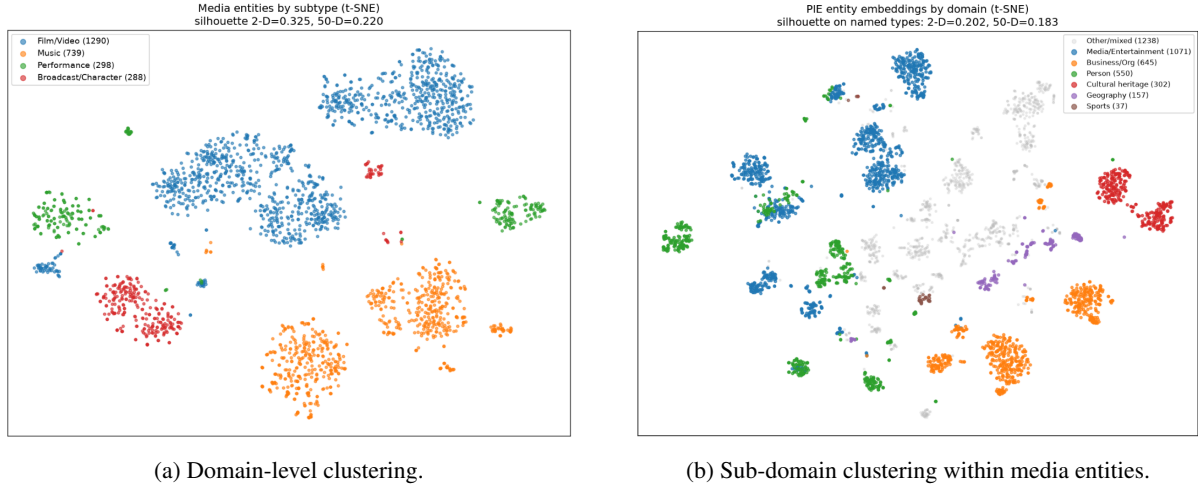


Figure 5: t-SNE visualization of PIE’s pooled entity embeddings  $\mathbf{v}_e$ . (a) Cultural-heritage and business entities form well-separated clusters, while person and media entities partially overlap. (b) Within the media domain, embeddings further separate by film/video, music, performance, and broadcast/character sub-types.

| Routing rule                             | Flagged    | % of test     | Err. caught | Flag prec.   | Err. rec.    | Acc.          | F1            |
|------------------------------------------|------------|---------------|-------------|--------------|--------------|---------------|---------------|
| Baseline (no review)                     | —          | —             | —           | —            | —            | 0.8742        | 0.8855        |
| Decision–score disagreement (DRSD)       | 667        | 13.34%        | 139         | 0.208        | 0.221        | 0.9020        | 0.9093        |
| Random (matched budget, 500-seed avg.)   | 667        | 13.34%        | 84.6        | 0.127        | 0.135        | 0.8911        | 0.8997        |
| Score near threshold ( $0.60 \pm 0.05$ ) | 30         | 0.60%         | 8           | 0.267        | 0.013        | 0.8758        | 0.8868        |
| PIE near threshold ( $0.50 \pm 0.05$ )   | 168        | 3.36%         | 46          | 0.274        | 0.073        | 0.8834        | 0.8931        |
| <b>PIE–DRSD disagreement (ours)</b>      | <b>775</b> | <b>15.50%</b> | <b>354</b>  | <b>0.457</b> | <b>0.563</b> | <b>0.9450</b> | <b>0.9470</b> |

Table 13: Comparison of human-review routing rules on the 5,000-pair evaluation set (decision threshold  $t=0.60$ ). “Errors caught” counts flagged pairs that were originally misclassified by DRSD; “Flag prec.” is the fraction of flagged pairs that were errors, and “Error rec.” is the fraction of DRSD’s 629 base errors that are flagged. Flagging pairs where PIE and DRSD disagree on the decision catches substantially more errors at a higher precision than routing on DRSD’s own score–decision disagreement, thresholding either signal near its decision boundary, or a random baseline matched to the same review budget (the random baseline’s best single seed of 500 reaches 111 errors caught, 0.8964 Acc, and 0.9043 F1).

## K Prompt & Output Examples

### (A) Gold-Label Prompt (GOLD\_TEMPLATE)

You will receive two JSON descriptions, and each description is generated from an entity's subgraph (attributes and relations).

You will ALSO receive the manually verified gold (ground-truth) decision label.

Your task is to generate:

- 1) A well-calibrated similarity score in [0, 1]
- 2) A high-quality justification that is CONSISTENT with the given decision and the score that you generate

This output will be used for knowledge distillation.

Your task has TWO phases:

#### **PHASE 1 (internal reasoning, not fully exposed):**

- Use the entity instance type that the model infers to determine which features are CORE vs. AUXILIARY.
- Compare the two entities with type-appropriate core attributes and relations.
- Identify matches that provide support, explainable discrepancies, conflicts, and information that is missing.
- Choose a similarity score that:
  - Is consistent with the evidence strength
  - Construct reasoning that supports BOTH the decision and the chosen score.

#### **IMPORTANT:**

- You MUST NOT contradict the gold decision that the prompt provides.
- The score that you generate MUST be consistent with the decision.
- Treat the gold decision as correct, even if the evidence is weak or ambiguous.

#### **PHASE 2 (final output only):**

- Output ONLY the simplified JSON that the template below defines.
- Do NOT include internal reasoning or mention the gold decision explicitly.

#### **[Entity A Description]**

{A\_TEXT}

#### **[Entity B Description]**

{B\_TEXT}

#### **[Gold Label]**

Decision: {GOLD\_DECISION}

#### **FINAL OUTPUT FORMAT (STRICT):**

Return ONLY valid JSON in the following format, with no additional text.

```
{
  "justification": [
    "Brief reason 1 (most important type-specific evidence or explanation)",
    "Brief reason 2",
    "Brief reason 3"
  ],
  "decision": "{GOLD_DECISION}",
  "score": 0.0
}
```

#### **Constraints:**

- The decision MUST exactly match the gold decision that the prompt provides.
- The score MUST be a float in [0, 1] with at most two decimal places.
- Justifications must be:
  - Type-appropriate
  - Consistent with both the decision and the score magnitude
- Do NOT add or remove fields.

Figure 6: Gold-label prompt used to generate a label-consistent decision and justification for distillation.

### (B) Score Prompt (SCORE\_TEMPLATE)

You will receive two JSON descriptions, and each description is generated from an entity's subgraph (attributes and relations).

Your task has TWO phases:

#### PHASE 1 (internal reasoning, not fully exposed):

- Compare the two entities with the most type-appropriate core attributes and relations.
- Identify high-confidence matches, matches that provide support, conflicts (high/medium/low), and information that is missing or uncertain.
- conflicts should be weighted by their importance to the entity type.
- Apply the scoring principles below to compute a similarity score between 0 and 1.

#### PHASE 2 (final output only):

- Output ONLY the simplified JSON that the template below defines.
- Do NOT include the full internal analysis in the final answer.
- Unique identifiers > entity type > type-specific core attributes > key relations > auxiliary attributes.
- Type mismatch or unique ID conflicts should strongly reduce the score.
- Missing information increases uncertainty but is not a conflict.

#### [Entity A Description]

{A\_TEXT}

#### [Entity B Description]

{B\_TEXT}

#### FINAL OUTPUT FORMAT (STRICT):

Return ONLY valid JSON in the following format, with no additional text.

```
{
  "justification": [
    "Brief reason 1 (most important type-specific evidence or conflict)",
    "Brief reason 2",
    "Brief reason 3"
  ],
  "score": 0.0
}
```

#### Constraints:

- The score must be a float in [0, 1] with at most two decimal places.
- Justification items must reflect features that are appropriate for the entity type that the model infers.

Figure 7: Score prompt used to estimate an evidence-based similarity score for an entity pair.

### (C) SFT Prompt (SFT\_TEMPLATE)

You will receive two JSON descriptions, and each description is generated from an entity's subgraph (attributes and relations).

Determine whether the two entities refer to the SAME entity or DIFFERENT entities.

#### [Entity A Description]

{A\_TEXT}

#### [Entity B Description]

{B\_TEXT}

#### Output:

- decision: SAME or DIFFERENT
- score: a similarity score in [0, 1]
- three brief justification points

Return only valid JSON in the specified format.

Figure 8: SFT prompt used as the student-model input at training and inference time.

#### (A) Gold-Label Output (JSON)

```
{
  "justification": [
    "Both entities share the title \"Memories\" but have different release years (2023 vs 2019).",
    "The listed composers and performers differ completely (J.Fla and multiple Western composers vs Maroon 5 and Korean-named composers).",
    "Additional metadata such as record label, language, and related works are unrelated. This indicates that they are distinct musical works."
  ],
  "decision": "DIFFERENT",
  "score": 0.22
}
```

Figure 9: Example gold-label prompt output containing a decision, score, and justification.

#### (B) Score Prompt Output (JSON)

```
{
  "justification": [
    "Both entities share the exact name \"Memories\" and overlapping core composers (Adam Levine and Jonathan/Jon Bellion). This indicates the same musical work.",
    "Entity A is a 2023 cover by J.Fla, whereas Entity B is the original 2019 recording by Maroon 5; the performer and release-date differences reflect different recordings, not a conflict on composition identity.",
    "Other attributes (record label, part_of album, instance_of) differ but are performance-specific and do not outweigh the strong composer and title match."
  ],
  "score": 0.78
}
```

Figure 10: Example score prompt output containing an evidence-based score and justification.

#### (C) SFT Target (JSON)

```
{
  "justification": [
    "Both entities share the title \"Memories\" but have different release years (2023 vs 2019).",
    "The listed composers and performers differ completely (J.Fla and multiple Western composers vs Maroon 5 and Korean-named composers).",
    "Additional metadata such as record label, language, and related works are unrelated. This indicates that they are distinct musical works."
  ],
  "decision": "DIFFERENT",
  "score": 0.78
}
```

Figure 11: Example SFT target combining the gold-label decision and justification with the score prompt score.