

All Relations at Once: Efficient Open-World Tabular Relation Discovery with Hierarchy-Aware Reward Shaping

Xinyu Zhang¹, Aibo Song^{1*}, Jingyi Qiu^{2†}, Tianbo Zhang¹

¹School of Computer Science and Engineering, Southeast University, Nanjing, China

²School of Artificial Intelligence and Information Technology,
Nanjing University of Chinese Medicine

{xyseu,absong,tianbozhang}@seu.edu.cn, jyqiu@njucm.edu.cn

Abstract

Understanding tabular semantics is essential for intelligent data management, yet data lake tables often lack explicit annotations, hindering interpretation of their structure and relations. Column Relation Extraction (RE), which identifies semantic relations between table columns, is a key step toward schema matching, data integration, and knowledge construction. Existing closed-set methods rely on predefined relation labels, while LLM-based generation offers greater flexibility but suffers from semantic ambiguity, coarse supervision, and inefficient pair-wise inference.

We propose a table-specific reinforcement learning framework for open-world RE. Instead of forcing predictions into a fixed label space, our method trains LLMs to generate free-form relations under fine-grained semantic feedback. We introduce a hierarchy-aware reward that converts semantic proximity into graded supervision, allowing the model to learn from near-miss predictions rather than treating all errors equally. We further propose relation-level policy optimization to improve efficiency: the model predicts all column-pair relations in one structured response, and each relation-level reward is assigned only to its corresponding token span. This reduces repeated table encoding and aligns optimization with relation-level correctness. Experiments demonstrate consistent improvements over strong LLM and table-specific baselines, especially on unseen relations and domains, highlighting the robustness and practicality for real-world table interpretation.

1 Introduction

Tables are a common and effective way to organize structured data in real-world scenarios. A central problem in table interpretation is Column Relation Extraction (RE), which aims to **identify**

*Corresponding author.

†Corresponding author.

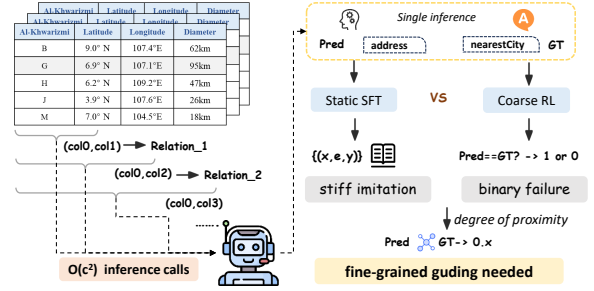


Figure 1: Drawbacks of current table RE paradigm.

semantic relationships between columns that reflect the underlying data logic (Zhang et al., 2019; Deng et al., 2022; Suhara et al., 2022; Zhang et al., 2025b). Accurate semantic relation discovery serves as a critical bridge between raw tabular data and structured knowledge, enabling schema alignment, knowledge graph construction, supporting downstream applications such as cross-source data fusion (Shigarov, 2023; Zhang et al., 2024).

However, real-world web tables make RE challenging due to heterogeneous schemas, missing annotations, and open-ended relation spaces. Semantically identical relations may appear under different column names across domains, rendering closed-set or rule-based methods brittle in open-domain settings, especially in open-domain settings where relation is diversity and evolving.

Recent advances in large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Bai et al., 2023) enable open-ended semantic inference, allowing relation descriptions to be generated without fixed label inventories. However, applying LLMs to RE remains challenging. Zero-shot prompting often yields lexically plausible yet semantically inconsistent outputs, suffering from synonym drift and hallucination. While supervised fine-tuning (SFT) alleviates this to some extent, it relies on hard labels, treating all incorrect predictions equally and failing to provide informative feedback for semantically similar outputs. This

limitation is particularly critical in semantic relation learning, where many predictions are only partially mismatched (e.g., *CityOfBirth* vs. *BirthPlace*). Moreover, although column-pair annotation is often treated as independent pair-wise classification, this is inefficient for generative LLMs, as it requires separate inference for each column pair within the same table, resulting in a quadratic number ($O(C^2)$) of inference calls.

To address these limitations, we propose a reinforcement learning (RL) framework for open-world RE that provides fine-grained supervision beyond binary correctness. Unlike SFT, RL enables the model to explore diverse relations and optimize generation behavior under task-specific reward signals, making it naturally suited for open-ended semantic prediction, as shown in Fig. 1.

We first design a **hierarchy-aware reward** that leverages a multi-level semantic taxonomy constructed from diverse relation schemas. Going beyond binary signals, this mechanism calibrates feedback via a continuous semantic manifold. Transition from rigid classification to graded credit assignment effectively mitigates the gradient sparsity of near-miss predictions, enabling smoother optimization and improved generalization.

We further propose **relation-level policy optimization**, a span-aware GRPO variant designed for structured RE generation. The model predicts all column-pair relations in one response, while relation-level rewards are back-propagated only to corresponding token spans. This enables efficient table-level generation while resolving the credit-assignment mismatch between sequence-level rewards and relation-level correctness.

Overall, our framework preserves the open-world nature of RE: the model generates relations freely without predefined candidates, while training is guided by graded semantic rewards and relation-level credit assignment. This design provides fine-grained supervision for near-miss predictions, concentrates optimization on relation tokens that determine correctness, and supports efficient all-pair generation within a single response. Our main contributions are as follows:

- We recast RE as an open-world generative task and introduce RL-based optimization to overcome the inherent bottlenecks of closed-set paradigms.
- We propose a hierarchy-aware reward that provides graded semantic feedback from a multi-level relation taxonomy, enabling informative learning from semantically close predictions.

- We introduce relation-level policy optimization, which predicts all column-pair relations in one response and assigns rewards to the corresponding relation spans for fine-grained credit assignment.
- Experiments demonstrate that our method significantly outperforms baselines, particularly exhibiting superior extrapolation capabilities in out-of-distribution scenarios and unseen relational types.

2 Related Work

Column Relation Extraction Table interpretation (Shigarov, 2023; Hulsebos et al., 2021; Wang et al., 2021) aims to interpret meaningful information from tabular data. Among its core components, RE plays a crucial role in identifying semantic relationships between table columns.

Recent approaches leverage PLMs (Devlin et al., 2019; Liu et al., 2019; Trabelsi et al., 2022; Yin et al., 2020), mainly falling into two categories: tabular embedding methods (e.g., TURL (Deng et al., 2022)) that learn cell representations for relation prediction, and fine-tuning approaches (e.g., DODUO (Suhara et al., 2022)) that use table serialization and multi-task learning. Above methods are inherently limited by predefined relations, restricting ability to generalize beyond closed relation spaces and handle open-domain scenarios with diverse and evolving relation types.

Large Language Models for Table Understanding With the rise of generative LLMs (Achiam et al., 2023; Bai et al., 2023; Brown et al., 2020; Touvron et al., 2023), recent studies have explored prompt-based methods (Kayali et al., 2023; Korini and Bizer, 2023) and fine-tuning for table tasks (Feuer et al., 2023; Korini and Bizer, 2024). TableLlama (Zhang et al., 2024) fine-tunes LLaMA-7B on multiple table tasks as an open-source generalist model, while MMTU (Xing et al., 2025) benchmarks expert-level table understanding and reasoning.

More recently, reinforcement learning has been applied to tabular reasoning. OpenTable-R1 (Qiu, 2025) trains an agent to interact with table retrieval and SQL execution tools for open-domain table question answering. Table-R1 (Yang et al., 2025b) employs reinforcement learning with verifiable rewards and task-specific verifiers to improve reasoning and inference-time scaling across table QA and fact-verification tasks. TableGPT-R1 (Yang et al., 2025a) further combines task-adaptive terminal verification, process-level reward shaping, and

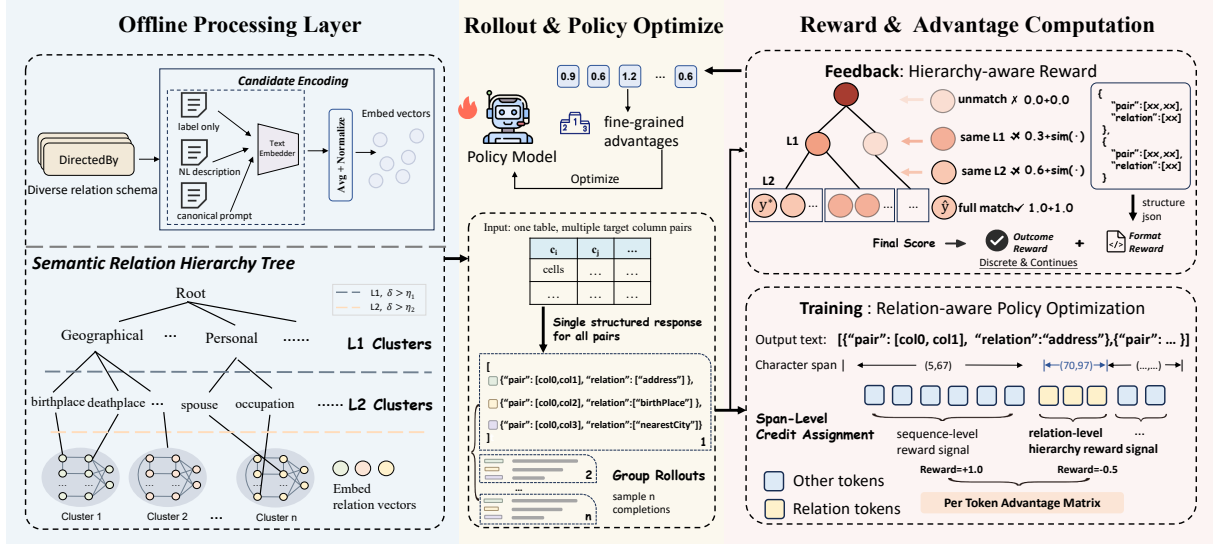


Figure 2: Overview of our framework.

multi-stage specialization for general agentic table reasoning.

Unlike these methods, our work targets open-world column relation generation, where the model must generate multiple relations without a predefined schema and semantically close predictions require graded supervision. We therefore introduce hierarchy-aware semantic rewards and relation-level token-span credit assignment, enabling open-ended relation generation while providing fine-grained credit to individual predictions within a structured response.

3 Methodology

3.1 Method Overview

This section presents an RL-based framework for open-world RE. The key challenge is that correctness is not strictly binary: many predictions may be lexically different but semantically close to the ground truth. Meanwhile, structured RE generation contains multiple relation predictions in a single response, making sequence-level reward assignment too coarse for effective optimization.

Fig. 2 shows the overall workflow. First, we construct a semantic hierarchy over relation schemas by encoding multi-source relation phrases and clustering them into coarse-to-fine groups. This hierarchy turns semantic proximity into graded reward signals. Second, the model performs all-pair generation, predicting all target column-pair relations of a table in one structured response, which avoids repeated pair-wise prompting. Third, we optimize the policy with relation-level credit assignment: each predicted relation is parsed into a token span,

and its individual reward is assigned only to that span, while non-relation tokens retain the sequence-level signal. This design preserves the efficiency of table-level generation while aligning the optimization signal with relation-level correctness.

3.2 Problem Statement

All-pair Generation To better fit the structure of table relation extraction, we adopt a creatively all-pair generation strategy, where the model predicts all target column-pair relations in a single response. This amortizes table understanding across column pairs while preserving the open-ended generation ability of LLMs. Given a table T contains a set of target column pairs $\mathcal{P}_T = \{p_1, \dots, p_M\}$, the policy model is required to generate all relations in one structured response:

$$y = [\{p_1, \hat{r}_1\}, \{p_2, \hat{r}_2\}, \dots, \{p_M, \hat{r}_M\}]. \quad (1)$$

This all-pair generation scheme amortizes table understanding across multiple column pairs and reduces repeated table encoding and avoids a quadratic number of LLM calls. Different column-pair predictions are naturally conditioned on a shared table context.

Unlike closed-set RE, the model generates free-form relations without relying on predefined label sets. A diverse schema set $\mathcal{R}$ is introduced only during reward computation phase of training as semantic anchors to evaluate prediction quality.

3.3 Hierarchy-Aware Reward

Reference Encoding. We first collected a set of schema phrases from diverse domains (e.g., DBpedia, Schema.org), exposing the model to varied

schema structures without restricting it to any fixed label set. Then we convert every reference relation phrase into a meaningful vector so that semantic distances reflect their real-world similarity.

For each reference relation $r \in \mathcal{R}$, we define three templates—phrase only (e.g., countryOfBirth), natural language description (e.g., the country where the person was born), and short canonical prompt (e.g., country of birth of the person)—to capture semantic variations and enhance robustness to wording variation. Each template is encoded by a embedding model $f_\phi(\cdot)$, and the relation’s final representation is their mean:

$$\mathbf{e}_r = \frac{1}{m} \sum_{i=1}^m f_\phi(t_i), \quad \bar{\mathbf{e}}_r = \frac{\mathbf{e}_r}{\|\mathbf{e}_r\|_2}, \quad (2)$$

with $m = 3$. Semantic similarity between relations is measured by cosine similarity $\text{sim}(r_i, r_j) = \bar{\mathbf{e}}_{r_i}^\top \bar{\mathbf{e}}_{r_j}$, and distance by $d(r_i, r_j) = 1 - \text{sim}(r_i, r_j)$.

We employ embedding models with well-calibrated similarity geometry and strong sensitivity to short phrase semantics, which are critical for fine-grained relation discrimination. The framework is encoder-agnostic and readily generalizes to any embedding model with comparable properties.

Hierarchy Construction. We construct a two-level semantic hierarchy over relation labels, grouping embeddings into fine-grained L2 clusters and coarser L1 groups. Formally, we learn a mapping $g : \mathcal{R} \rightarrow \{(L2, L1)\}$, where L2 captures closely related relations and L1 aggregates semantically similar L2 clusters.

To support our hierarchical reward design which requires consistent coarse-to-fine grouping under multiple distance thresholds, we adopt agglomerative clustering¹. Unlike flat clustering (e.g., k -means (MacQueen, 1967)), it produces a dendrogram that enables coherent multi-level partitioning via threshold cuts, ensuring that L1 and L2 clusters are derived consistently.

Specifically, we compute the pairwise distance matrix $D_{i,j} = d(r_i, r_j)$ and perform agglomerative clustering to obtain a linkage tree. We then select two thresholds $0 < t_{L2} < t_{L1} < 1$, cutting the dendrogram at t_{L2} to form L2 clusters. For each cluster C_k , we compute its centroid:

$$\mathbf{c}_k = \frac{1}{|C_k|} \sum_{r \in C_k} \bar{\mathbf{e}}_r, \quad \bar{\mathbf{c}}_k = \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|_2}. \quad (3)$$

¹scikit-learn: AgglomerativeClustering

We then cluster the L2 centroids using the same procedure and cut at t_{L1} to obtain L1 assignments. To improve stability, we enforce a minimum cluster size $N_{\min} = 3$, merging smaller clusters into their nearest neighbors.

Reward Computation We convert the model predictions into a scalar training reward that (i) assigns graded credit to semantically close predictions and (ii) enforces the expected response format. Predictions matching the ground truth get the highest reward; those in the same L2 cluster receive significant partial credit; those in the same L1 (different L2) get minor credit; unrelated predictions receive no credit. Let y^* denotes the ground truth and $\hat{y}$ is the model prediction. We define the discrete hierarchical distance:

$$h(\hat{y}, y^*) = \begin{cases} 0, & \hat{y} = y^* \\ 1, & \hat{y}, y^* \in \text{Same L2} \\ 2, & \hat{y}, y^* \in \text{Same L1} \\ 3, & \text{otherwise} \end{cases} \quad (4)$$

We first define a **Discrete tier-reward mapping** $R_D: \{0, 1, 2, 3\} \rightarrow \mathbb{R}$ with $R_D(0) = r_0 > R_D(1) = r_1 > R_D(2) = r_2 > R_D(3) = r_3$, which are set in our implementation as $(r_0, r_1, r_2, r_3) = (1.0, 0.6, 0.3, 0.0)$. To give finer credit within a tier, we combine the tier reward with a similarity-based **Continuous reward** within $[0, 1]$:

$$R_C(\hat{y}, y^*) = \begin{cases} 1, & h(\hat{y}, y^*) = 0 \\ \text{sim}(\bar{\mathbf{e}}_{\hat{y}}, \bar{\mathbf{e}}_{y^*}), & h(\hat{y}, y^*) \in \{1, 2\} \\ 0, & h(\hat{y}, y^*) = 3 \end{cases} \quad (5)$$

The final reward is:

$$r = \begin{cases} \lambda R_D(h) + (1-\lambda) R_C, & \text{if format valid} \\ -\beta_{\text{fmt}}, & \text{otherwise} \end{cases} \quad (6)$$

where $\lambda \in [0, 1]$ controls emphasis on discrete tiers and continuous similarity. Semantic reward is strictly bounded in $[0, 1]$ and the format penalty $\beta_{\text{fmt}} = 0.2$ is applied only to malformed outputs.

3.4 Relation-Aware Policy Optimization

Sequence-Level Advantage Assignment. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as the backbone training algorithm. For each table, we sample G completions $\{o_i\}_{i=1}^G$ from π_{old} (policy at the time of generation). Each completion is parsed into a set of relation objects, where an object is considered *valid* if it is parseable JSON with both pair and non-empty relation fields. The sequence reward

aggregates per-relation rewards together with a penalty for hallucinated column pairs:

$$R_i^{\text{sem}} = \sum_{k=1}^{m_i} r(\hat{y}_{i,k}, y_{i,k}^*), \quad R_i^{\text{struct}} = \gamma \cdot |\hat{\mathcal{P}}_i \setminus \mathcal{P}^*|, \quad (7)$$

where m_i is the number of valid relations in o_i , $\hat{\mathcal{P}}_i$ and $\mathcal{P}^*$ are the predicted and target column-pair sets, and $\gamma = -0.2$ penalizes spurious predictions. The sequence-level advantage is then

$$\hat{A}_i = \frac{R_i^{\text{struct}} - \mu_R}{\sigma_R + \epsilon}, \quad (8)$$

Span-Level Advantage Assignment. A key limitation of standard GRPO is that the sequence-level advantage $\hat{A}_i$ is broadcast uniformly to every generated token, conflating the quality of structured predictions with that of surrounding reasoning and formatting tokens. This is particularly harmful for RE, where a single completion contains multiple independently-assessable column-pair predictions, each localized to a distinct token span.

We address this with a *per-span advantage override*. During rollout, the reward function returns, for each completion o_i , both the scalar total reward R_i (Eq. 8) and a set of disjoint span-reward pairs $\{([s_{i,k}, e_{i,k}], r_{i,k})\}_{k=1}^{m_i}$, where $[s_{i,k}, e_{i,k}]$ delimits the k -th relation prediction $\hat{y}_{i,k}$ and $r_{i,k} = r(\hat{y}_{i,k}, y_{i,k}^*)$ is its per-relation reward from Eq. (6). Span boundaries are recovered by incrementally decoding the token sequence and tracking character offsets, ensuring exact alignment with model’s own token ids. Span rewards are normalized within each column pair across G rollouts of same prompt:

$$\tilde{r}_{i,k} = \frac{r_{i,k} - \mu_{p(k)}}{\sigma_{p(k)} + \epsilon}, \quad (9)$$

where $p(k)$ denotes the column pair predicted by the k -th span, avoiding cross-pair variance contamination. $\hat{A}_i$ and $\tilde{r}_{i,k}$ stem from **disjoint** structural/semantic rewards, producing non-overlapping advantages. The final token-level advantage is then

$$A_i(t) = \begin{cases} \tilde{r}_{i,k}, & \text{if } \exists k \text{ s.t. } t \in [s_{i,k}, e_{i,k}], \\ \hat{A}_i, & \text{otherwise.} \end{cases} \quad (10)$$

Relation tokens receive fine-grained supervision from their own prediction quality, while tokens outside any span (e.g., structural delimiters) retain the sequence-level signal $\hat{A}_i$. Both rewards are standardized within groups and concentrates gradient updates on tokens whose content directly determines correctness, while the reasoning scaffold is still shaped by overall output quality.

Policy Update. Given the token-level advantage $A_i(t)$, the policy is updated with objective:

$$\mathcal{L}(\theta) = -\frac{1}{|B|} \sum_{i \in B} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left[\min(\rho_{i,t} A_i(t), \text{clip}(\rho_{i,t}, 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}) A_i(t)) - \beta \delta_{\text{KL}}^{(i)}(t) \right]. \quad (11)$$

Here $o_{i,t}$ denotes the t -th token of completion o_i , and $|o_i|$ is its number of non-padding tokens. β controls the KL regularization strength against a frozen reference policy.

Instead of issuing $O(C^2)$ pair-wise prompts, our strategy predicts all column-pair relations in one structured generation, reducing repeated table serialization and inference calls. Meanwhile, relation-level reward decomposition and span-aware advantage assignment provide independent supervision for each predicted relation, so partially correct outputs remain informative and rewards are applied only to the corresponding relation tokens. This combines efficiency of all-pair annotation with fine-grained credit assignment, making RL practical for structured table relation extraction rather than a generic response-level optimization.

4 Experiments

4.1 Datasets

We employ two representative datasets to evaluate our method. SemTab 2019 (Jiménez-Ruiz et al., 2020) contains four rounds (R1-R4) of challenge data, with DBpedia as target KG. SOTAB V2² contains CPA labels covering Schema.org types. More details in Appendix A.

4.2 Baselines

We consider baselines of two main genres: powerful **Closed-source LLMs**: GPT-5.1³, GPT-4.1-mini⁴, Gemini-25-pro⁵, DeepSeek-R1⁶, Seed-1.6⁷, which are the latest powerful general model. **Table-Specific Models**: OpenTableR1 (Qiu, 2025), TableLlama (Zhang et al., 2024), TableGPT-2 (Su et al., 2024), STaR (Zhang et al., 2026), StructLM (Zhuang et al., 2024) and TableLLM (Zhang et al., 2025a). which has been trained for tabular tasks including RE.

²<https://webdatacommons.org/structureddata/sotab/v2/>

³<https://developers.openai.com/api/docs/models/gpt-5.1>

⁴<https://openai.com/index/gpt-4-1/>

⁵<https://deepmind.google/models/gemini/pro/>

⁶<https://github.com/deepseek-ai/DeepSeek-R1>

⁷<https://seed.bytedance.com/en/seed1-6>

Table 1: Main results of baselines. Best in **bold**, second best underlined.

Base Model	Method	SemTab-R4			SOTAB-V2		
		Precision	Recall	Micro-F1	Precision	Recall	Micro-F1
<i>Closed-Source Models</i>							
GPT-5.1	Few-Shot	32.72	32.86	32.79	70.18	72.84	71.48
GPT-4.1-mini	Few-Shot	5.44	22.56	8.77	44.04	62.86	51.80
Gemini-25-pro	Few-Shot	44.17	30.18	35.86	74.93	71.85	73.36
DeepSeek-R1	Few-Shot	29.26	28.88	29.07	66.13	64.36	65.23
Doubao-Seed-1.6	Few-Shot	31.65	31.03	31.34	64.44	68.38	66.35
<i>Table-Specific Models</i>							
OpenTableR1-4B	Few-Shot	8.2	19.51	11.55	19.96	16.75	18.21
TableLlama-7B	Few-Shot	5.56	1.21	1.99	11.07	10.25	10.64
TableGPT2-7B	Few-Shot	12.54	12.97	12.75	30.82	33.25	31.99
STaR-8B	Few-Shot	17.63	9.24	12.12	53.93	36.92	43.84
StructLM-13B	Few-Shot	4.1	19.5	6.78	16.75	15.13	15.90
TableLLM-13B	Few-Shot	10.86	14.32	12.35	28.47	31.65	29.98
Llama-3.1-8B-Ins	Zero-Shot	17.68	11.72	14.10	14.60	23.97	18.15
	SFT	21.04	20.78	20.91	48.62	52.34	50.41
	Our RL	35.10	28.45	31.43	68.43	62.86	65.53
Qwen2.5-14B-Ins	Zero-Shot	18.44	18.47	18.46	37.31	45.09	40.83
	SFT	35.82	30.27	32.81	68.95	65.83	67.35
	Our RL	45.23	34.46	39.12	71.84	73.62	72.72

4.3 Implementation Details

We employ the Llama-3.1-8B-Instruct (Touvron et al., 2023) and Qwen2.5-Instruct-14B (Team, 2024) as base models and conduct comprehensive experiments with LSRL (Liang, 2025) and vLLM (Kwon et al., 2023). We adopt Qwen3-Embedding-0.6B (Team, 2025) as the encoder throughout. We perform full-parameter GRPO training directly on our base models, with rollout number set to 8, lr set to $5e-6$, and β set to 0.001. For fair comparison, we include SFT baselines for the same backbones, allowing us to isolate the effect of RL beyond SFT. For two genres of baselines, we adopt the N-shot paradigm, with N defaultly set to 3. More details in Appendix A.

4.4 Main Results

Tab. 1 reports the performance of all methods.

Table-specific baselines struggle with open-world RE. Existing table-specific models perform poorly across both datasets, with the best baseline (STaR-8B) reaching only 12.12 F1 on SemTab-R4 and 43.84 F1 on SOTAB-V2. This confirms that closed-set fine-tuning fails to generalize to the open-ended relation space, where lexical variation and unseen schemas dominate.

Our RL method substantially outperforms open-source baselines and rivals closed-source LLMs. Built on Qwen2.5-14B-Ins, our RL framework achieves **39.12** F1 on SemTab-R4 and **72.72** F1 on SOTAB-V2, surpassing the best table-

specific baseline by **+26.37** and **+28.88** F1, respectively. Notably, it outperforms Gemini-2.5-pro on SemTab-R4 and consistently exceeds GPT-5.1, DeepSeek-R1, and Doubao-Seed-1.6 across both benchmarks, demonstrating that fine-grained semantic feedback effectively unlocks open-world relation discovery capabilities.

Consistent gains across model scales validate the framework. The Zero-Shot $\rightarrow$ SFT $\rightarrow$ Our RL trajectory yields monotonic improvements on both backbones: SFT alone provides limited gains due to its rigid imitation of hard labels, whereas our RL delivers substantially larger gains (+17.33 F1 on Llama, +20.66 F1 on Qwen for SemTab). The same trend indicates that hierarchy-aware reward shaping benefits even smaller backbones. These results jointly verify that graded semantic supervision and relation-level optimization are the key drivers of performance, rather than model scale alone.

4.5 Analysis of Open-Set Predictions

In-Schema Evaluation. To better assess the ability to recognize known relations, we restrict evaluation to predictions that fall within the GT label space in Tab. 2. Under this setting, performance improves substantially for LLM-based methods (e.g., +13–17 Micro-F1 across two base models), while gains on table-specific models are markedly smaller. This suggests that a large portion of performance loss is not due to incorrect understanding, but because they produce valid relations outside the

Table 2: Open-Set evaluation on SemTab-R4. Δ_1 denotes the improvement over Main F1. Δ_2 denotes the improvement over In-Schema F1.

Base Model	Method	Main F1	In-Schema F1	Δ_1	LLM-Judge	Δ_2
OpenTableR1-4B	Few-Shot	11.55	18.43	+6.88	24.16	+5.73
TableLlama-7B	Few-Shot	1.99	5.27	+3.28	8.94	+3.67
TableGPT2-7B	Few-Shot	12.75	24.86	+12.11	31.52	+6.66
STaR-8B	Few-Shot	12.12	23.94	+11.82	29.78	+5.84
StructLM-13B	Few-Shot	6.78	11.62	+4.84	16.85	+5.23
TableLLM-13B	Few-Shot	12.35	22.71	+10.36	28.43	+5.72
<i>Ours</i>						
Llama-3.1-8B-Ins	Zero-Shot	14.10	27.83	+13.73	35.16	+7.33
	SFT	20.91	35.42	+14.51	42.78	+7.36
	Our-RL	31.43	45.86	+14.43	64.92	+19.06
Qwen2.5-14B-Ins	Zero-Shot	18.46	34.78	+16.32	42.36	+7.58
	SFT	32.81	47.95	+15.14	55.24	+7.29
	Our-RL	39.12	54.71	+15.59	73.58	+18.87

Table 3: Ablation study results. #Calls denotes the number of LLM calls per table with n column pairs.

Variant	SemTab F1	SOTAB F1	#Calls per Table
Our-RL (full)	39.12	72.72	$O(1)$
w/o Hierarchy Reward	31.85	65.43	$O(1)$
w/o Span-Level Credit	34.27	67.96	$O(1)$
w/o All-pair Generation	35.46	67.18	$O(n)$

predefined label set. In contrast, table-specific models trained on closed schemas rarely produce such “creative” predictions—their errors are genuine semantic failures rather than schema mismatches.

Further inspection shows that many such predictions are semantically appropriate alternatives rather than true errors. For instance, the model may generate *cinematographer* for movie-production relations, which is semantically valid but not included in the annotation schema. This highlights a key limitation of closed-set evaluation. In contrast, our framework encourages such behavior by rewarding semantic similarity rather than exact matches. These results indicate that our model learns a flexible semantic space. In dynamic open-world scenarios with evolving knowledge, enabling new meaningful relation discovery supports more intelligent, scalable knowledge acquisition.

LLM-as-Judge Evaluation. To further assess open-world semantic correctness, we adopt an LLM-as-Judge protocol (Liu and et al., 2023; Zheng and et al., 2023), where GPT-5.1 determines whether each prediction is semantically equivalent or acceptable. Tab. 2 shows that LLM-Judge Acc scores are consistently higher than In-Schema F1,

Table 4: Generalization performance results.

Base Model	Paradigm	SemTab-R1		
		P	R	F1
OpenTableR1-4B	Zero-Shot	34.12	33.05	33.58
TableLlama-7B	Zero-Shot	19.58	12.11	14.96
TableGPT2-7B	Zero-Shot	31.90	32.46	32.18
STaR-8B	Zero-Shot	31.82	30.17	30.97
StructLM-13B	Zero-Shot	20.57	16.93	18.57
TableLLM-13B	Zero-Shot	22.74	20.18	21.38
<i>Ours</i>				
Llama-3.1-8B-Ins	Zero-Shot	20.13	25.86	22.64
	SFT	11.43	18.18	14.04
	Our RL	37.02	36.41	36.71
<i>Ours</i>				
Qwen2.5-14B-Ins	Zero-Shot	22.41	21.09	21.73
	SFT	14.81	17.72	16.13
	Our RL	42.06	38.79	40.36

confirming that many predictions excluded by exact matching are in fact semantically valid.

Crucially, the gap between LLM-Judge and In-Schema F1 differs sharply across paradigms: Zero-Shot and SFT methods exhibit only modest gaps (+5 ~ 8 points), whereas Our-RL achieves substantially larger gaps (+18 points on both backbones). This indicates that hard-label SFT primarily memorizes schema vocabulary—its remaining errors are largely true errors—while our hierarchy-aware reward enables the model to populate the semantic neighborhood around correct answers, producing predictions that closed-set metrics penalize but LLM judges recognize as valid. This validates our central claim: graded semantic supervision unlocks open-ended generalization that exact-match training fundamentally cannot achieve.

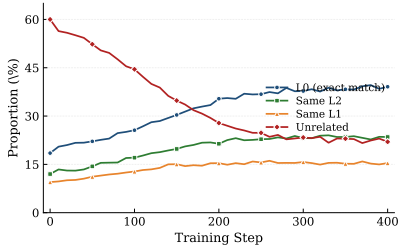


Figure 3: Evolution of prediction tier distribution.

4.6 Ablation Study

We ablate three key components of our framework on Qwen2.5-14B-Ins; results are shown in Tab. 3.

Hierarchy-aware reward is the most critical component. Replacing graded hierarchical rewards with binary 0/1 correctness causes the largest drop (-7.27 F1 on SemTab-R4, -7.29 on SOTAB-V2), confirming that fine-grained semantic feedback rather than rigid exact-match supervision—is the primary driver of open-world performance.

Span-level credit assignment is essential for structured generation. Reverting to standard GRPO degrades F1 by 4.85/4.76 points, showing that uniformly broadcasting a single reward to all tokens conflates relation tokens with reasoning scaffolds and dilutes the learning signal.

All-pair generation improves both accuracy and efficiency. Switching to pair-wise inference causes a substantial F1 drop ($-3.66/-5.54$) and inflates inference cost from $O(1)$ to $O(n)$ calls per table. The gain arises from joint reasoning over all column pairs in a shared context, preserving dependencies fragmented by pair-wise decoding. Combined with the $n \times$ reduction in LLM calls, all-pair generation is particularly suitable for wide tables.

4.7 Unseen Relation Generalization

To evaluate generalization to unseen relations, we perform zero-shot transfer test across all models on SemTab-R1, whose label space has minimal overlap with the training set, simulating an open-world setting. As shown in Tab. 4, Our-RL achieves **40.36** F1 on Qwen2.5-14B and **36.71** on Llama-3.1-8B, surpassing the best table-specific baseline (OpenTableR1-4B, 33.58) by $+6.78$ and $+3.13$ F1 respectively. Strikingly, SFT *degrades* performance on both backbones (-8.60 and -5.60 F1 vs. Zero-Shot), revealing that hard-label imitation overfits to the training schema and harms transfer, whereas our graded semantic supervision learns relation structure that generalizes to unseen labels.

Table 5: Inference efficiency comparison on SemTab.

Setting	Paradigm	#Calls per Table	Tokens (K)	Latency (s)
R4	Pair-wise	4	2.31	13.4
	All-pair	1	0.81	5.2
R1	Pair-wise	5	6.68	27.8
	All-pair	1	1.73	8.5
Speedup		—	2.9–4×	2.6–3.3×

4.8 Training Dynamics Analysis

To understand how hierarchy-aware rewards drive learning, we track the distribution of prediction tiers throughout training on SemTab-R4 (Qwen2.5-14B-Ins). As shown in Fig. 3, unrelated predictions shrink dramatically from 60.0% to 22.0%, while exact matches (L0) nearly double from 18.5% to 39.1%. Notably, near-miss tiers (L1, L2) expand earliest and peak around step 200, indicating that the model first migrates predictions into the correct semantic neighborhood before crystallizing them into exact matches. This staged convergence—*unrelated* $\rightarrow$ *semantically close* $\rightarrow$ *exact*—is precisely the trajectory binary rewards cannot induce: without graded credit, near-miss predictions provide no learning signal.

4.9 Efficiency of All-pair Generation

We compare pair-wise vs. all-pair inference on Qwen2.5-14B-Ins. Table 5 shows that all-pair generation consistently reduces both token consumption and wall-clock latency, with the gap widening as tables grow: on small SemTab-R4 tables we observe $2.9 \times$ token reduction and $2.6 \times$ speedup, while on R1 tables with more columns the savings reach $4 \times$. This validates the $O(n) \rightarrow O(1)$ complexity reduction per table, making our method deployable for large-scale table interpretation.

5 Conclusion

We present a RL framework for open-world RE that addresses two long-standing bottlenecks: rigid binary supervision and quadratic pair-wise inference. Our hierarchy-aware reward converts semantic proximity into graded feedback, while relation-level policy optimization aligns credit assignment with structured all-pair generation. Experiments across in-distribution, out-of-distribution, and open-set settings show consistent gains over both LLM and table-specific baselines.

6 Limitations

While our method demonstrates strong performance across in-distribution, out-of-distribution, and open-set settings, several aspects remain open for future exploration. Compared to standard SFT, reinforcement learning with group rollouts naturally introduces additional training overhead, as multiple completions must be sampled per prompt. While our all-pair generation strategy substantially reduces per-rollout cost, training still requires more compute than supervised alternatives. Exploring sample-efficient variants such as offline RL, replay buffers, or partial-rollout reuse is a promising direction.

Acknowledgments

This work is supported by Funded by Frontier Technologies R&D Program of Jiangsu (BF2025076).

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.
- Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.
- Benjamin Feuer, Yurong Liu, Chinmay Hegde, and Juliana Freire. 2023. Archetype: A novel framework for open-source column type annotation using large language models. In *Proceedings of VLDB*.
- Madelon Hulsebos, Sneha Gathani, James Gale, Isil Dillig, Paul Groth, and Çağatay Demiralp. 2021. Making table understanding work in practice. *arXiv preprint arXiv:2109.05173*.
- Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. 2020. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings*.
- Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2023. Chorus: foundation models for unified data discovery and exploration. *arXiv preprint arXiv:2306.09610*.
- Keti Korini and Christian Bizer. 2023. Column type annotation using chatgpt. In *Proceedings of VLDB*.
- Keti Korini and Christian Bizer. 2024. Column property annotation using large language models. In *Proceedings of ESWC*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of SIGOPS*.
- Jiaqing Liang. 2025. Lsrl: Memory efficient large model training framework. <https://github.com/ldefine/lssl>.
- Yang Liu and et al. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *EMNLP*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.
- Zipeng Qiu. 2025. Opentable-r1: A reinforcement learning augmented tool agent for open-domain table question answering. *ArXiv*, abs/2507.03018.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Alexey Shigarov. 2023. Table understanding: Problem overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(1):e1482.
- Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, et al. 2024. Tablegpt2: A large multimodal model with tabular data integration. *arXiv preprint arXiv:2411.02059*.
- Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *Proceedings of SIGMOD*.

- Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Qwen Team. 2025. Qwen3 technical report. *arXiv preprint, arXiv:2505.09388*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Mohamed Trabelsi, Zhiyu Chen, Shuo Zhang, Brian D Davison, and Jeff Hefflin. 2022. Strubert: Structure-aware bert for table search and matching. In *Proceedings of the ACM Web Conference 2022*, pages 442–451.
- Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.
- Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Lingjiao Chen, Dongmei Zhang, Surajit Chaudhuri, and HV Jagadish. 2025. Mmtu: A massive multi-task table understanding and reasoning benchmark. *arXiv preprint arXiv:2506.05587*.
- Saisai Yang, Qingyi Huang, Jing Yuan, Liangyu Zha, Kai Tang, Yuhang Yang, Ning Wang, Yucheng Wei, Liyao Li, Wentao Ye, Hao Chen, Tao Zhang, Junlin Zhou, Haobo Wang, Gang Chen, and Junbo Zhao. 2025a. TableGPT-R1: Advancing tabular reasoning through reinforcement learning. *arXiv preprint arXiv:2512.20312*.
- Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. 2025b. Table-R1: Inference-time scaling for table reasoning. *arXiv preprint arXiv:2505.23621*.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.
- Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. 2019. Sato: Contextual semantic type detection in tables. *arXiv preprint arXiv:1911.06311*.
- Huajian Zhang, Mingyue Cheng, Yucong Luo, and Xiaoyu Tao. 2026. Star: Towards effective and stable table reasoning via slow-thinking large language models. In *Proceedings of the ACM Web Conference 2026, WWW '26*, page 7189–7200, New York, NY, USA. Association for Computing Machinery.
- Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. Tablellama: Towards open large generalist models for tables. In *Proceedings of NAACL*.
- Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Yang Li, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, Jifan Yu, Shu Zhao, Juanzi Li, and Jie Tang. 2025a. TableLLM: Enabling tabular data manipulation by LLMs in real office usage scenarios. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10315–10344, Vienna, Austria. Association for Computational Linguistics.
- Xinyu Zhang, Aibo Song, Jingyi Qiu, Jiahui Jin, Tianbo Zhang, and Xiaolin Fang. 2025b. Exploring multi-modal relation extraction of hierarchical tabular data with multi-task learning. In *Proceedings of ACL*.
- Lianmin Zheng and et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*.
- Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W. Huang, Jie Fu, Xiang Yue, and Wenhui Chen. 2024. Structlm: Towards building generalist models for structured knowledge grounding. *ArXiv, abs/2402.16671*.

A Detailed Datasets Information

The method is evaluated on two widely used english benchmarks of structured web tables: SemTab2019⁸ and SOTAB V2⁹. SemTab2019 focuses on table-to-knowledge-graph matching and semantic annotation tasks. Given CSV tables as input, the goal is to align table elements with entities, types, or relations in a knowledge graph, and 80/20 train-test split is used. SOTAB V2 is a large-scale benchmark for semantic annotation of real-world web tables. It is constructed through extensive cleaning and filtering, resulting in high-quality annotated samples that cover a wide range of Schema.org types and diverse table scenarios. Due to its large raw size, we sample 8k instances for experiments and perform row truncation when necessary to meet input length constraints. Statistics are summarized in Tables 6 and 7.

For fair comparison, we include SFT baselines for the same backbones to isolate the effect of RL beyond SFT. We conduct SFT with LoRA. For SemTab2019, we use lr of 1e-4, batch size of 4, epoch of 4, and a LoRA rank r=8 for Llama-3.1-8B-Ins, r=16 for Qwen2.5-14B-Ins. For SOTAB, we use lr of 2e-5, batch size of 2, epoch of 2, and a LoRA rank r=8 for Llama-3.1-8B-Ins, r=16 for Qwen2.5-14B-Ins. All experiments were conducted within 4*80G H100.

⁸<https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2019/index.html>

⁹<https://webdatacommons.org/structureddata/sotab/v2/>

Table 6: Statistics of SemTab.

Metric	Round 1	Round 4
# Tables	64	817
Avg. # Rows	142 ± 139	63 ± 52
Avg. # Columns	5 ± 2	4 ± 1
Avg. # Cells	696 ± 715	268 ± 223
# Target Column Pairs	116	2747

Table 7: Statistics of SOTAB V2.

Metric	Total	Train	Test
# Tables	29,723	29,158	565
# Column Pairs	112,334	109,994	2,340
# Websites	55,511	—	—

B Process of Hierarchy Construction

The process of performing hierarchical clustering on the label set used for training is outlined in the Algorithm. 1.

Algorithm 1 Clustering and Hierarchy Construction

Input: Relation set $\mathcal{R}$, encoder $f_\phi(\cdot)$, templates $\{t_1, \dots, t_m\}$, thresholds t_{L2}, t_{L1} s.t. $0 < t_{L2} < t_{L1} < 1$, minimum cluster size $N_{\min}$.

Output: Mapping $g : \mathcal{R} \rightarrow \{(L2_{id}, L1_{id})\}$

- 1: Reference encoding:
- 2: **for all** $r \in \mathcal{R}$ **do**
- 3: **for** $i \leftarrow 1$ to m **do**
- 4: $\mathbf{e}_i \leftarrow f_\phi(t_i)$
- 5: **end for**
- 6: $\mathbf{e}_r \leftarrow \frac{1}{m} \sum_{i=1}^m \mathbf{e}_i$
- 7: $\bar{\mathbf{e}}_r \leftarrow \mathbf{e}_r / \|\mathbf{e}_r\|_2$
- 8: **end for**
- 9: Compute similarity $\text{sim}(r_i, r_j) = \bar{\mathbf{e}}_{r_i}^\top \bar{\mathbf{e}}_{r_j}$ and distance matrix $D_{i,j} = 1 - \text{sim}(r_i, r_j)$
- 10: **Stage I: form L2 (fine) clusters**
- 11: Perform agglomerative clustering on D (linkage) and cut dendrogram at height t_{L2}
- 12: Obtain L2 clusters $\{C_k\}$
- 13: Compute L2 centroids and merge small clusters:
- 14: **for all** C_k **do**
- 15: $\mathbf{c}_k \leftarrow \frac{1}{|C_k|} \sum_{r \in C_k} \bar{\mathbf{e}}_r$
- 16: $\bar{\mathbf{c}}_k \leftarrow \mathbf{c}_k / \|\mathbf{c}_k\|_2$
- 17: **end for**
- 18: **for all** C_k with $|C_k| < N_{\min}$ **do**
- 19: Find nearest L2 centroid $\bar{\mathbf{c}}_{k'}$ (min distance)
- 20: Merge C_k into $C_{k'}$
- 21: Recompute $\bar{\mathbf{c}}_{k'}$
- 22: **end for**
- 23: **Stage II: form L1 (coarse) groups**
- 24: Compute pairwise distances between L2 centroids $\{\bar{\mathbf{c}}_k\}$
- 25: Perform agglomerative clustering on centroids and cut at height t_{L1}
- 26: Assign each L2 cluster to an L1 group
- 27: **return** mapping $g(r) = (L2_{id}(r), L1_{id}(r))$

Table 8: Sensitivity analysis of the discrete-continuous trade-off coefficient λ

λ	Precision	Recall	F1
0.0 (continuous only)	41.83	31.74	36.10
0.2	43.27	32.86	37.36
0.4	44.58	33.91	38.52
0.6 (default)	45.23	34.46	39.12
0.8	44.71	33.92	38.58
1.0 (discrete only)	42.95	32.18	36.79

C Sensitivity Analysis

We analyze the trade-off between discrete tier rewards and continuous similarity in Eq. 6 by sweeping $\lambda \in [0, 1]$ on SemTab-R4. Tab. 8 reveals an inverted-U pattern peaking at $\lambda=0.6$ (F1 = 39.12), with both extremes degrading performance. Pure continuous similarity ($\lambda=0$) yields the lowest F1 (36.10) because the reward landscape becomes overly smooth, failing to enforce sharp distinctions between tiers; pure discrete tiers ($\lambda=1$) drop to 36.79 as intra-tier predictions receive identical credit, losing fine-grained discrimination. Importantly, performance remains stable across $\lambda \in [0.4, 0.8]$ (F1 within 0.6 points), demonstrating that our method is robust to this hyperparameter and does not require careful tuning.

D Reward Hacking Analysis

Any reward signal richer than binary correctness opens potential avenues for the policy to maximize reward without genuine task improvement. We argue that our method’s design closes the most plausible attack surfaces by construction, and provide empirical diagnostics to verify that no such hacking behavior emerges in practice.

D.1 Structural Defenses

Three structural choices jointly prevent the policy from exploiting our reward:

- **Two-component reward** (discrete tier + continuous similarity, Eq. 6) requires *both* cluster membership and semantic proximity. Generic “safe-word” predictions (e.g., *relation*, *property*) yield low similarity scores within their tier, collapsing the partial credit they might otherwise harvest.
- **Gated format reward** Unlike additive format rewards that can dominate the gradient when the semantic reward is zero (i.e., tier-3 predictions), our gated formulation guarantees that

format compliance acts as a hard precondition rather than an independent reward channel. Within the format-valid subspace, the per-rollout reward and advantages differences are determined exclusively by the hierarchy-aware semantic component. This closes the most plausible attack surface where a policy could maximize format reward while ignoring semantic correctness.

- **Span-level credit assignment** (Eq. 10) scores each predicted relation independently against its own ground truth, eliminating any benefit from repeating the same plausible relation across all column pairs.

D.2 Empirical Diagnostics

Setup. To verify that no hacking behavior emerges in practice, we compute three diagnostic metrics on rollouts from SemTab-R4, comparing our hierarchy-aware reward against a binary-reward variant and an SFT baseline that contains no reward signal. We sample rollouts at training steps {200, 300, 400} and report the average to avoid cherry-picking a single checkpoint; per-step variation is within ± 0.3 across all metrics. The “Generic phrase rate” is computed against a pre-defined list of 12 schema-agnostic placeholder phrases (e.g., *relation*, *property*, *related to*, *associated with*) curated independently of model predictions and verified to have zero overlap with the ground-truth label vocabulary, eliminating definitional circularity.

Table 9: Reward-hacking diagnostics on SemTab-R4. “Generic Rate” is the fraction of predictions falling into a pre-defined list of 12 schema-agnostic placeholder phrases; “Mode Concentration” is the share of the top-1 most-predicted relation.

Diagnostic	Binary	SFT	Ours	GT
Prediction entropy (bits) $\uparrow$	3.91	4.53	4.82	4.97
Generic phrase rate (%) $\downarrow$	14.6	4.8	2.3	—
Mode concentration (%) $\downarrow$	18.7	9.2	6.4	5.9

Distributional alignment with ground truth.

Our method’s predicted relation distribution closely tracks the ground truth across all metrics: prediction entropy (4.82 vs. 4.97 bits) and mode concentration (6.4% vs. 5.9%) are nearly identical. In contrast, the binary-reward variant exhibits a characteristic failure pattern: the model converges on a small set of generic relations and re-emits them across many column pairs within the same table.

This single mechanism explains both the entropy drop (3.91, -0.91 bits vs. ours) and the inflated output length—repeated emission of the same safe word both concentrates the distribution and lengthens the output. The elevated generic phrase rate (14.6% vs. 2.3%) corroborates this interpretation. The SFT baseline sits between the two RL variants on every metric, consistent with hard-label imitation memorizing the label vocabulary without the distributional smoothing that graded reward provides.

These diagnostics indicate that our method’s outputs are *distributionally aligned* with the ground-truth relation distribution along all three behavioral axes we measure, whereas binary supervision drifts toward repetitive generic emissions. We note that distributional alignment is a necessary but not sufficient condition for the absence of reward hacking—a more direct test would compare reward magnitude against task-level accuracy across training. The substantial gap between binary and hierarchy-aware variants on *both* reward dynamics and downstream F1 (+7.27, Tab. 3) provides this complementary evidence: under our reward, higher reward consistently translates to higher task performance, reflecting alignment between the verification signal and genuine task quality rather than surface-form exploitation.