# POLI-RL: A POINT-TO-LIST REINFORCEMENT LEARNING FRAMEWORK FOR CONDITIONAL SEMANTIC TEXTUAL SIMILARITY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Conditional Semantic Textual Similarity (C-STS) measures the semantic proximity between text segments under a specific condition, thereby overcoming the ambiguity inherent in traditional STS. However, existing methods are largely confined to discriminative models, failing to fully integrate recent breakthroughs in the NLP community concerning Large Language Models (LLMs) and Reinforcement Learning (RL). RL is a particularly well-suited paradigm for this task, as it can directly optimize the non-differentiable Spearman ranking metric and guide the reasoning process required by C-STS. However, we find that naively applying listwise RL fails to produce meaningful improvements, as the model is overwhelmed by a complex, coarse-grained reward signal. To address this challenge, we introduce PoLi-RL, a novel Point-to-List Reinforcement Learning framework. PoLi-RL employs a two-stage curriculum: it first trains the model with simple pointwise rewards to establish fundamental scoring capabilities, then transitions to a hybrid reward that combines pointwise, pairwise, and listwise objectives to refine the model's ability to discern subtle semantic distinctions. Crucially, we propose an innovative Parallel Slice Ranking Reward (PSRR) mechanism that computes ranking rewards in parallel slices, where each slice comprises same-indexed completions from different samples. This provides a precise, differentiated learning signal for each individual completion, enabling granular credit assignment and effective optimization. On the official C-STS benchmark, PoLi-RL achieves a Spearman correlation coefficient of 48.18, establishing a new SOTA for the cross-encoder architecture. As the first work to successfully apply RL to C-STS, our study introduces a powerful and effective paradigm for training LLMs on complex, ranking-based conditional judgment tasks. Our code and checkpoints are available at `https://anonymous.4open.science/r/PoLi-RL`.

## 1 INTRODUCTION

As a core research area in Computational Linguistics, Semantic Textual Similarity (STS) (Agirre et al., 2013) finds extensive applications across diverse scenarios, including topic modeling, dialogue systems, text summarization, and agent memory (Tang et al., 2025). However, traditional STS tasks exhibit inherent ambiguity because similarity definitions are often susceptible to observer bias. To address this limitation, the Conditional Semantic Textual Similarity (C-STS) task was developed (Deshpande et al., 2023). By incorporating an explicit natural language condition, C-STS enables more precise and objective similarity judgments, yet simultaneously imposes higher demands on a model's reasoning capabilities. For instance, consider the following two text fragments: "A player is shooting from beyond the three-point line" and "A player is taking a free throw". Under the condition "The activity of the player", their similarity is high. However, under the condition "The player's distance from the basket", their similarity is low.

Research on this nascent task has yielded three primary paradigms: Bi-encoder (Liu et al., 2025), Tri-encoder (Lin et al., 2024), and Cross-encoder (Li et al., 2024). The Cross-encoder architecture, which processes the text pair and the guiding condition simultaneously, is the most compatible with modern generative pre-trained models. Despite this, the integration of C-STS with LLMs remains in its early stages. Current LLM applications are limited to two main approaches: direct inference via few-shot prompting, where even state-of-the-art models struggle to achieve satisfactory results (Deshpande et al., 2023); and their use as feature extractors for generating text embeddings (Yamada & Zhang, 2025), which is an extension of the discriminative paradigm. To the best of our knowledge, no prior work has applied an end-to-end LLM-based cross-encoder to the C-STS task, nor has any integrated it with advanced training techniques like reinforcement learning (RL), leaving a significant research gap.

This paper aims to fill this gap. We posit that incorporating RL into an LLM-based cross-encoder paradigm is a natural fit. This is reflected in two aspects: First, C-STS requires sophisticated, scenario-based reasoning. For example, in the basketball case described earlier, to correctly assess similarity under the "distance" condition, the model must move beyond surface-level semantics to identify the underlying spatial relationship between 'beyond the three-point line' and 'at the free throw line', a process demanding strong abstraction and inference. RL, through its explicit reward signals, can more effectively guide the reasoning process of LLMs (Guo et al., 2025). Second, from an optimization standpoint, RL aligns closely with the task's evaluation criteria. The Spearman correlation coefficient (Zar, 2005), a core evaluation metric of C-STS, is a non-differentiable measure of ranking quality. Traditional SFT methods can only indirectly and approximately optimize this objective through loss functions like Mean Squared Error (MSE) (Zhang & Li, 2024b). In contrast, RL allows for the direct optimization of ranking-based reward functions that are designed to correlate strongly with the final Spearman metric, maintaining a higher degree of consistency with the final evaluation.

However, a naive application of RL to this task presents significant challenges. As illustrated in Figure 1, our preliminary experiments indicate that directly applying a single listwise ranking reward (e.g., Spearman's correlation coefficient) across an entire batch of completions does not show any advantages compared to the few-shot method. This approach suffers from two fundamental problems. First, the ranking objective is too complex for a model that has not yet learned the task's fundamental scoring rules, often causing the training to collapse. Second, a single reward computed across the entire batch is too coarse to provide precise credit assignment, as a few poor completions can unfairly penalize other good ones.

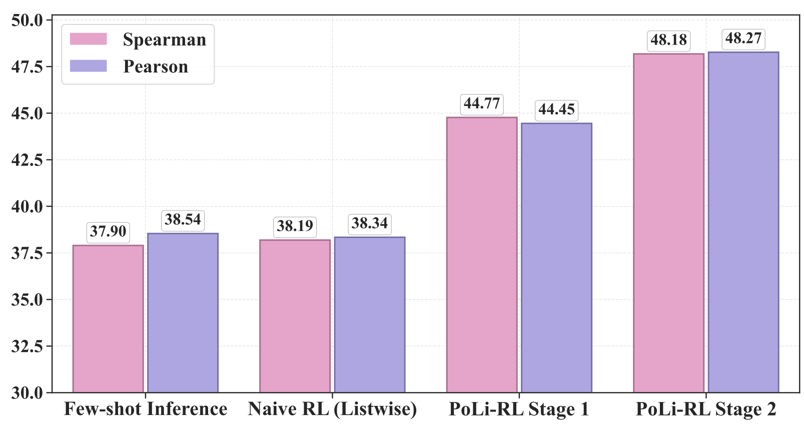


Figure 1: Performance comparison of different strategies on the C-STS task. Directly applying listwise ranking rewards for RL does not significantly outperform the few-shot baseline. In contrast, both stages of our method (PoLi-RL) achieve substantial improvements, validating its effectiveness.

To address these challenges, we propose **PoLi-RL**, a two-stage **Po**int-to-**Li**st **R**einforcement **L**earning framework. PoLi-RL features a two-stage curriculum to manage the complexity of the learning task. In the first stage, we use simple pointwise rewards to ground the model in the basic scoring rules of the task. Building on this foundation, the second stage introduces a richer, hybrid reward signal that combines a stable pointwise anchor with more nuanced pairwise and listwise ranking rewards. This progressive approach refines the model's ability to discern subtle semantic differences while ensuring stable and effective training.

Furthermore, to resolve the problem of a coarse-grained reward signals that arises from ranking all completions in a batch together, we innovatively introduce a Parallel Slice Ranking Reward (PSRR) mechanism, which utilizes a two-level decomposition. First, for a batch of input samples, the model generates $G$ completions for each. We then form $G$ "parallel slices", where the $i$-th slice is composed of the $i$-th completion from every sample. Second, and more importantly, within each slice, rather than assigning a single reward, we compute the rank difference for each individual completion against its ideal rank. This two-level decomposition allows each of the $N \times G$ completions to receive a unique and precise reward that reflects its quality, thereby enabling granular credit assignment and stable training.

The primary contributions of this paper are outlined as follows:

- To the best of our knowledge, this is the first work to propose an end-to-end, LLM-based cross-encoder for the C-STS task and the first to employ reinforcement learning for training in this domain.
- We design and implement PoLi-RL, a novel two-stage training curriculum that mitigates the instability of direct rank-based optimization by progressing from a simple pointwise reward to a more complex hybrid reward.
- We propose the Parallel Slice Ranking Reward (PSRR) mechanism, which delivers precise and differentiated learning signals by computing ranking rewards within independent 'parallel slices.' This mechanism offers a generalizable strategy for other ranking and retrieval tasks involving multiple generation candidates.
- On the official C-STS benchmark, PoLi-RL achieves a Spearman's correlation coefficient of 48.18, establishing a new SOTA for the cross-encoder architecture and surpassing strong closed-source models, including GPT-4 (43.6) (Achiam et al., 2023). Our qualitative analysis further reveals our method's advantages in understanding complex conditions.

## 2 Methodology

This section details our proposed strategy. We begin by formulating the C-STS task within an end-to-end, LLM-based cross-encoder paradigm in subsection 2.1. Then, in subsection 2.2, we map this task onto the mathematical framework of Reinforcement Learning and specifiy its optimization objective. Finally, in subsection 2.3, we provide a comprehensive description of our PoLi-RL framework, including its two-stage design and the innovative PSRR mechanism.

### 2.1 Problem Formulation

The core objective of C-STS is to learn a scoring function that accurately reflects the semantic similarity between two text segments under a specific condition. Formally, each C-STS data sample is defined as a tuple $x = (t_1, t_2, c, y)$, where $t_1, t_2$ are two text segments, $c$ is the natural language condition, and $y \in [1, 5]$ is the human-annotated similarity judgement on the Likert scale (Likert, 1932). Notably, the label $y$ corresponds to a fine-grained set of semantic criteria. According to the C-STS annotation guidelines, the meanings of the scores are as follows: (1) Completely dissimilar; (2) Thematically related but dissimilar; (3) Roughly equivalent, but with some important information differences; (4) Mostly equivalent, with some unimportant details differing; (5) Completely equivalent. This level of granularity demands that the model perform fine-grained reasoning beyond surface-level semantics, posing a significant challenge to its capabilities.

A unique characteristic of the C-STS dataset is its paired structure: samples are organized in adjacent pairs that share the same text segments $(t_1, t_2)$ but feature different conditions and maintain a deterministic ordinal relationship between their labels, i.e., $y_{\text{high}} \geq y_{\text{low}}$. This structure provides a solid foundation for our pairwise reward design, as detailed in subsection 2.3.

Our task is to train a scoring model $\pi_\theta$, parameterized by $\theta$. For each sample $x$, the model takes a unified prompt $p = [\mathcal{I}, \mathcal{E}, x]$ (detailed in Appendix A.2) as input, comprising the instruction $\mathcal{I}$, $K$ few-shot demonstrations $\mathcal{E} = \{(x_k, y_k)\}_{k=1}^{K}$, and the data sample $x$, to generate an output sequence $o = \pi_\theta(p)$. From this sequence, we parse the final predicted score, $\tilde{y} = \text{Parse}(o)$. The overall training objective is to maximize the ranking consistency between the set of predicted scores $\{\tilde{y}_i\}_{i=1}^{N}$ and the ground-truth scores $\{y_i\}_{i=1}^{N}$ by optimizing the policy $\pi_\theta$, a process primarily measured by Spearman's correlation coefficient. Since this metric is rank-based and non-differentiable, RL emerges as a more promising optimization paradigm than traditional supervised fine-tuning.

### 2.2 Reinforcement Learning for C-STS

We formulate the C-STS task as a Markov Decision Process (MDP), defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where the agent is the LLM policy $\pi_\theta$. The generation process is modeled as a sequence of decisions, where each step involves generating a single token. A state $s_t \in \mathcal{S}$ at timestep $t$ is the sequence of tokens generated so far, conditioned on the initial prompt, i.e., $s_t = (p, o_{<t})$. An action $a_t \in \mathcal{A}$ is the selection of the next token $o_t$ from the model's vocabulary, governed by the policy $\pi_\theta(a_t|s_t)$, which provides a probability distribution over all possible tokens. The transition function $\mathcal{T}$ is deterministic, where the next state $s_{t+1}$ is formed by appending the selected token $a_t$ to $s_t$. We employ a terminal reward setting, where a reward $\mathcal{R}_T = \mathcal{R}(x, o)$ is given only after the entire sequence $o$ has been generated. Finally, the discount factor $\gamma$ is set to 1 to ensure that the terminal reward is backpropagated without decay to all actions that contributed to the final output. Based on this framework, the objective is to find the optimal parameters $\theta^*$ that maximize the expected reward over the data distribution $\mathcal{D}$:

$$\theta^* = \arg\max_\theta \mathbb{E}_{x \sim \mathcal{D}, o \sim \pi_\theta(p)}[R(x, o)] \tag{1}$$

To optimize this objective, we employ Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025), an extension of GRPO (Shao et al., 2024) that introduces several key techniques for effective RL. For each sample $x$, the policy generates a set of $G$ completions $\{o_i\}_{i=1}^G$. A scalar reward $r_i = R(x, o_i)$ is computed for each completion. The advantage $\hat{A}_i$ for each completion is then calculated by normalizing its reward against the statistics of the entire group's rewards via Z-score normalization:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^G)}{\text{std}(\{r_i\}_{i=1}^G) + \epsilon} \tag{2}$$

These advantages are used to define the objective function for updating the model parameters $\theta$:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_\theta(\cdot|p)} \left[ \frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \left( \frac{\pi_\theta(o_{i,t}|p, o_{i,<t})}{[\pi_\theta(o_{i,t}|p, o_{i,<t})]_{\text{nograd}}} \hat{A}_{i,t} \right) \right] \tag{3}$$

### 2.3 PoLi-RL: A Two-Stage Reinforcement Learning Framework

As previously established, our optimization goal is to maximize the expected reward, making the design of the reward function $\mathcal{R}$ central to our method. To ensure stable and effective optimization, we propose PoLi-RL, a framework built upon a two-stage progressive reward curriculum. This subsection details our pipeline and its reward mechanisms, as illustrated in Figure 2.

**Stage I: Foundational Skill Acquisition.** The goal of Stage I is to ground the model in the fundamental scoring rules of the C-STS task. For each input sample, the policy generates $G$ completions, from which we parse a set of predicted scores $\{\tilde{y}_j\}_{j=1}^G$. The total reward for Stage I, $R_{S1}$, is a weighted sum of three components:

$$R_{S1} = \lambda_1 R_{\text{pointwise}} + \lambda_2 R_{\text{binary}} + \lambda_3 R_{\text{format}} \tag{4}$$

The Pointwise Accuracy Reward ($R_{\text{pointwise}}$) is the primary component in Stage I. It measures the normalized distance between the predicted score $\tilde{y}_j$ and the ground-truth score $y_j$.

$$R_{\text{pointwise}} = 1 - \frac{|\tilde{y}_j - y_j|}{\max(Y) - \min(Y)} \tag{5}$$

where $\max(Y) = 5$ and $\min(Y) = 1$ are the bounds of the label space.

To counter reward hacking, where the model tends to output safe intermediate scores, we introduce a Binary Judgement Reward ($R_{\text{binary}}$). According to the C-STS guideline that scores $\geq 3$ indicate similarity while scores $\leq 2$ indicate dissimilarity, this reward encourages the model to first master this basic binary classification:

$$R_{\text{binary}} = \begin{cases} 1 & \text{if } (\tilde{y}_j \geq 3 \wedge y_j \geq 3) \vee (\tilde{y}_j < 3 \wedge y_j < 3) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

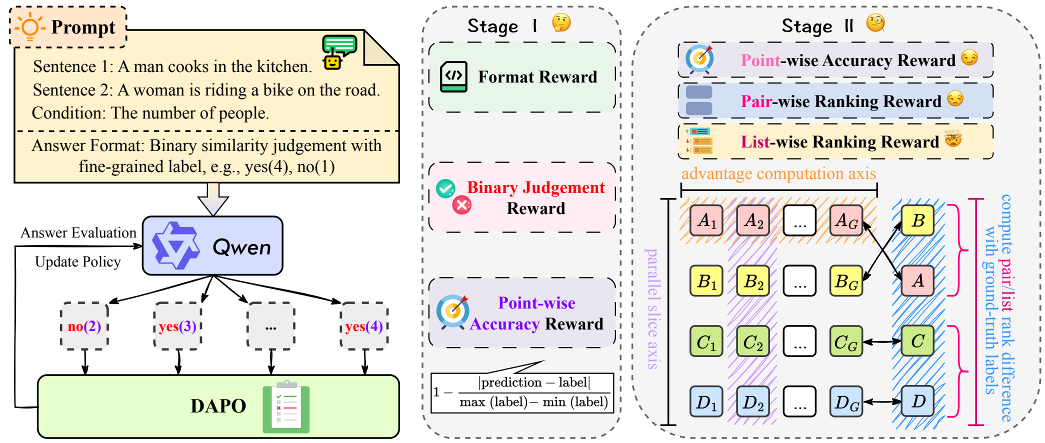


Figure 2: An overview of the PoLi-RL framework. The framework employs a two-stage curriculum, progressing from Stage I, where the model learns foundational scoring rules, to Stage II, which refines the model's ability to discern fine-grained semantic differences. The core of our method is the PSRR mechanism in Stage II, where pairwise and listwise ranking rewards are computed vertically within slices of same-indexed completions to provide precise, differentiated learning signals.

Finally, a simple Format Reward ($R_{\text{format}}$) ensures the output adheres to the required structure, which consists of a binary judgment ('yes' or 'no') followed by the numerical score in parentheses.

**Stage II: Fine-Grained Semantic Refinement.** After the model has acquired basic scoring abilities in Stage I, Stage II refines its capacity to discern subtle semantic differences by introducing a richer, hybrid reward signal that incorporates both pairwise and listwise ranking objectives.

PSRR: A FINE-GRAINED REWARD MECHANISM. A primary challenge in directly optimizing ranking metrics is that a single, batch-wide reward signal is too coarse to assign credit precisely. To address this, we propose the Parallel Slice Ranking Reward (PSRR) mechanism. The core idea of PSRR is to restructure the generated outputs to enable more granular reward computation. For a batch of $N$ samples, we begin by having the policy generate $G$ completions $\{o_{i,1}, \dots, o_{i,G}\}$ for each sample $x_i$. From each completion $o_{i,j}$, we then parse the predicted score $\tilde{y}_{i,j}$. Instead of treating the completions as a single flat list, we organize these $N \times G$ predicted scores into $G$ "parallel slices". Each slice, denoted as $Y^j_{\text{pred}}$, is defined as the collection of the $j$-th predicted score from all $N$ samples in the batch: $Y^j_{\text{pred}} = \{\tilde{y}_{1,j}, \tilde{y}_{2,j}, \dots, \tilde{y}_{N,j}\}$, where $j \in \{1, \dots, G\}$. This slicing architecture is the foundation upon which our advanced ranking rewards are built, ensuring that each completion receives a specific learning signal based on its relative performance within its slice.

A sufficiently large slice size $N$ is crucial for a stable and meaningful ranking signal. To make this computationally feasible with limited GPU memory, we leverage gradient accumulation. Specifically, we first generate the full set of $N \times G$ completions and organize them into parallel slices to compute rewards and advantages globally. Subsequently, we process smaller sub-batches sequentially to compute losses and accumulate gradients over multiple backward passes before executing a single optimizer step. This strategy makes our reward design practically feasible, enabling the model to learn from a rich, large-scale ranking signal without requiring prohibitive memory.

PAIRWISE RANKING REWARD. Computed within each parallel slice $P_j$, this reward leverages the paired structure of the C-STS dataset to provide a local ranking signal. It is applied only to adjacent input samples $(x_i, x_{i+1})$ that form a valid pair. For such a pair, we define the predicted difference as $\Delta_{\text{pred}} = \tilde{y}_{i,j} - \tilde{y}_{i+1,j}$ and the true difference as $\Delta_{\text{true}} = y_i - y_{i+1}$. The reward $R^{\text{pairwise}}_{i,j}$ is then a piecewise function:

$$R^{\text{pairwise}}_{i,j} = \begin{cases} 0 & \text{if } \operatorname{sgn}(\Delta_{\text{pred}}) \neq \operatorname{sgn}(\Delta_{\text{true}}) \\ R_{\text{base}} + (1 - R_{\text{base}}) \cdot \left(1 - \frac{|\Delta_{\text{pred}} - \Delta_{\text{true}}|}{\text{max\_error}}\right) & \text{if } \operatorname{sgn}(\Delta_{\text{pred}}) = \operatorname{sgn}(\Delta_{\text{true}}) \end{cases} \tag{7}$$

This function first checks if the basic ranking preference is correct using the sign function $\text{sgn}(\cdot)$. If the order is wrong, the reward is zero. If correct, a base reward $R_{\text{base}}$ is given, plus an additional reward that measures the normalized distance between the predicted and true score differences. Here, max_error is the maximum possible score difference, which is 3 for paired samples.

LISTWISE RANKING REWARD. While the pairwise reward focuses on local comparisons, the listwise reward provides a more global ranking perspective within each slice. It is calculated as the normalized difference between a completion's predicted rank within its slice and the ideal rank of its ground-truth label, formulated as:

$$R_{i,j}^{\text{listwise}} = 1 - \frac{|\text{Rank}(\tilde{y}_{i,j}, Y_{\text{pred}}^{j}) - \text{Rank}(y_i, Y_{\text{true}})|}{N-1} \tag{8}$$

where $Y_{\text{true}} = \{y_1, \dots, y_N\}$ is the set of true labels for the current batch, the function $\text{Rank}(v, S)$ returns the rank of element $v$ within the set $S$ (from 1 to $N$), and the division by $N-1$ normalizes the rank error to the range $[0, 1]$.

The final reward for Stage II, $R_{S2}$, combines the robust Pointwise Reward from Stage I as a stabilizing anchor with the new ranking-based rewards enabled by PSRR. The total reward is a weighted combination of these three components:

$$R_{S2} = \mu_1 R_{\text{pointwise}} + \mu_2 R_{\text{pairwise}} + \mu_3 R_{\text{listwise}} \tag{9}$$

# 3 EXPERIMENTS

To empirically validate the effectiveness of our proposed PoLi-RL framework, we conduct a comprehensive set of experiments. We begin by detailing our experimental setup in subsection 3.1, including the dataset, evaluation metrics, baselines, and implementation details. Following this, in subsection 3.2, we present the main results, comparing PoLi-RL against a suite of strong baselines. Finally, in subsection 3.3, we conduct a series of ablation studies to analyze the contributions of our framework's key components.

## 3.1 EXPERIMENTAL SETUP

We build PoLi-RL upon the Qwen3-8B (Yang et al., 2025) model using the *ms-swift* framework (Zhao et al., 2024) for RL training. All experiments are conducted on the official C-STS dataset (Deshpande et al., 2023). Following prior work, we use Spearman correlation as the primary metric and Pearson as the secondary. We compare our method against three baseline categories: First, the discriminative models in the cross-encoder setting. Second, powerful generative LLMs, such as Flan-UL (Tay et al., 2022), Flan-T5 (Chung et al., 2024) and Tk-Instruct (Wang et al., 2022). Finally, our own SFT and few-shot implementations on Qwen3-8B for direct comparison.

## 3.2 MAIN RESULTS

Table 1 summarizes the performance of our framework, PoLi-RL, which establishes a new state-of-the-art (SOTA) for the cross-encoder architecture with a Spearman correlation of **48.18**. The significance of this achievement is best understood through a series of key comparisons. First, PoLi-RL surpasses the previous cross-encoder SOTA, SEAVER, by a significant margin of **4.35** points. Second, the advantage of our method is particularly stark on the Qwen3-8B model, where it yields substantial absolute improvements of **10.28** points over few-shot inference and **7.76** points over standard SFT, showcasing the substantial benefits of our progressive, multi-component reward optimization.

More remarkably, the efficacy of our framework is further highlighted when benchmarked against vastly larger models. Our 8B parameter model not only substantially outperforms powerful proprietary models like GPT-4 but also demonstrates a commanding lead over other large open-source models like Flan-T5. This result illustrates that our RL-based method can cultivate nuanced reasoning capabilities in moderately-sized models, making them highly capable and competitive for complex conditional judgment tasks without relying on scale alone.

Table 1: Main results on the official C-STS benchmark. All scores are reported as Spearman/Pearson correlation coefficients multiplied by 100. Results marked with † are obtained from (Deshpande et al., 2023), while ‡ denotes results from (Li et al., 2024).

| Methods | Training Paradigm | Parameters | Spearman ↑ | Pearson ↑ |
|---|---|---|---|---|
| | | *Discriminative Model Baselines (Cross-Encoder Architecture)* | | |
| RoBERTa$_{\text{LARGE}}$† | SFT | 355M | 40.7 | 40.8 |
| SimCSE$_{\text{LARGE}}$† | SFT | 355M | 43.2 | 43.2 |
| SEAVER SimCSE$_{\text{LARGE}}$‡ | SFT | 355M | 43.83 | 43.81 |
| | | *Generative Large Language Model Baselines* | | |
| Flan-T5$_{\text{XXL}}$† | Few-shot | 11B | 30.6 | - |
| Flan-UL2† | Few-shot | 20B | 23.5 | - |
| Tk-Instruct† | Few-shot | 11B | 17.8 | - |
| GPT-3.5† | Few-shot | 175B | 15.3 | - |
| GPT-4† | Few-shot | - | 43.6 | - |
| | | *Our Implementation on Qwen3-8B* | | |
| Qwen3-8B | Few-shot | 8B | 37.9 | 38.54 |
| Qwen3-8B | SFT | 8B | 40.42 | 40.83 |
| PoLi-RL (Ours) | RL | 8B | **48.18** | **48.27** |

### 3.3 Ablation Studies

**Effectiveness of the Two-Stage Curriculum and Reward Components.** Table 2 dissects the effectiveness of our progressive training schedule. We first observe that a Naive RL approach (Row 2), which uses only a single, batch-wise listwise reward from scratch, yields negligible improvement over the few-shot baseline (Row 1), demonstrating the need for a more structured curriculum. Our PoLi-RL Stage I (Row 3) addresses this by building a robust foundation, substantially outperforming the few-shot baseline by 6.87 points. Within this stage, ablating the binary reward (Row 4) leads to a discernible dip in performance, validating its role in grounding the model in the task's basic binary judgment.

Building upon this, the full PoLi-RL model (Row 5) further boosts performance by another 3.41 points (Row 5). Deconstructing the success of this final stage reveals that both ranking signals are vital: removing the listwise reward (Row 6) incurs the most significant penalty, while removing the pairwise reward (Row 7) also hinders performance. These findings confirm that both the two-stage curriculum and each of its constituent reward signals are essential for achieving optimal results, with the listwise signal being the most critical component in the final refinement stage.

Table 2: Ablation study on PoLi-RL's two-stage training design and its reward components. The Δ column shows the absolute improvement in Spearman correlation over the indicated baseline.

| Method | Reward Component(s) | Spearman ↑ | Pearson ↑ | Δ (Spearman) |
|---|---|---|---|---|
| (1) Few-shot Inference | - | 37.9 | 38.54 | - |
| (2) Naive RL | Listwise | 38.19 | 38.34 | +0.29 vs. (1) |
| (3) PoLi-RL (Stage I) | Pointwise + Binary | 44.77 | 44.45 | **+6.87** vs. (1) |
| (4) - w/o Binary | Pointwise | 44.19 | 43.54 | -0.58 vs. (3) |
| (5) PoLi-RL (Full) | Pointwise + Pairwise + Listwise | **48.18** | **48.27** | **+3.41** vs. (3) |
| (6) - w/o Listwise | Pointwise + Pairwise | 46.71 | 46.37 | -1.47 vs. (5) |
| (7) - w/o Pairwise | Pointwise + Listwise | 47.6 | 47.59 | -0.58 vs. (5) |

**Sensitivity to Reward Weights in PoLi-RL Stage II.** Table 3 details our analysis of the framework's sensitivity to reward weights in Stage II. The results reveal that the framework is robust to weight variations, with peak performance achieved by moderately increasing the pairwise weight to 1.5. The model performs strongly even with the default equidistant weights (1:1:1), and shows considerable tolerance to pairwise signal, as halving its weight to 0.5 results in only a marginal performance drop to 47.77. Similarly, for the pointwise and listwise weights, deviations from their baseline of 1.0 result in only minor fluctuations. Crucially, despite these variations, the framework exhibits stable convergence across all configurations, exhibiting no training collapse. This confirms that our hybrid reward design effectively resolves the optimization difficulties encountered by naive listwise approaches.

Table 3: Ablation study on the reward weights ($\mu_1, \mu_2, \mu_3$) in PoLi-RL's Stage II.

| **Method** | $\mu_1$ **(Pointwise)** | $\mu_2$ **(Pairwise)** | $\mu_3$ **(Listwise)** | **Spearman** ↑ | **Pearson** ↑ |
|---|---|---|---|---|---|
| | 1.0 | 1.0 | 1.0 | 47.83 | 47.83 |
| | 1.0 | 1.5 | 1.0 | **48.18** | **48.27** |
| PoLi-RL (Stage II) | 1.5 | 1.0 | 1.0 | 47.3 | 47.23 |
| | 1.0 | 1.0 | 1.5 | 47.46 | 47.48 |
| | 1.0 | 0.5 | 1.0 | 47.77 | 47.31 |
| | 0.5 | 1.0 | 1.0 | 47.36 | 47.18 |
| | 1.0 | 1.0 | 0.5 | 47.39 | 47.27 |

**Sensitivity to Parallel-Slice Size.** To determine the optimal configuration for our PSRR mechanism, we study the impact of the slice size $N$, with results presented in Table 4. The empirical results reveal a clear trend: performance peaks at an intermediate slice size of $N = 24$ and degrades as the size deviates in either direction. This suggests that an optimal balance is required for the ranking signal. A slice that is too small may provide a less stable ranking signal, while one that is too large makes the ranking task overly complex for the model to learn effectively. This finding validates the design principle behind our PSRR mechanism: a carefully-sized, localized ranking signal is more effective than a purely global or an overly-restricted one.

Table 4: Analysis on the impact of the parallel slice size ($N$) in Stage II. $N$ represents the number of samples used for listwise ranking reward computation in each slice.

| **Method** | $N$ **(Slice Size)** | **Spearman** ↑ | **Pearson** ↑ |
|---|---|---|---|
| | 16 | 47.16 | 46.96 |
| | 24 | **48.18** | **48.27** |
| PoLi-RL(Stage II) | 32 | 47.44 | 47.19 |
| | 40 | 47.18 | 47.32 |
| | 48 | 46.78 | 46.84 |

# 4 Analysis

## 4.1 Analysis of Prediction Error Distribution

Figure 3 visualizes the distribution of absolute prediction errors on the C-STS validation set. The plot reveals several key insights into the models' behaviors. While an error of 1 is the most frequent outcome for all models, likely reflecting the inherent nuances of the C-STS scale, a clear progression of improvement is evident. Compared to the raw and SFT model, PoLi-RL demonstrates a more favorable error distribution. Firstly, it achieves the highest density of perfect predictions (error=0). More importantly, the density of the PoLi-RL curve in the high-error regions is consistently the lowest. This shows that our method significantly reduces the frequency of large, unreliable errors, yielding a more stable and reliable model.

## 4.2 Qualitative Analysis: A Case Study on Nuanced Reasoning

To qualitatively analyze our framework's nuanced reasoning ability, we present a case study in Figure 4 on the challenging condition "The person's connection with the ground". The baseline models

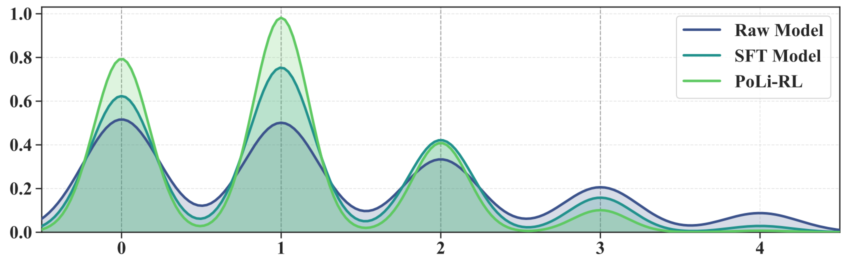


Figure 3: Distribution of absolute prediction errors for the raw model, SFT model, and PoLi-RL. The $x$-axis represents the absolute error ($|\text{Predicted} - \text{True}|$), and the $y$-axis is the probability density.

show clear deficiencies: the Few-shot model exhibits a brittle, literal interpretation, fixating on the preposition 'on' while ignoring the real-world context, leading to an incorrect prediction. The SFT Baseline, while achieving a closer score, shows uncertainty and fails to make the necessary fine-grained distinction. In contrast, PoLi-RL applies world knowledge to infer that both "jumping" and "doing a flip" result in the same "not connected" state, arriving at the accurate score of 4 by identifying the differing actions leading to the state as unimportant details. This case study demonstrates that PoLi-RL cultivates the contextual inference ability required for complex conditional judgments.

**Case Study: Model Comparison on a Reasoning Sample**

**Sentence 1:** A person riding a snowboard jumps high over the snowy hill.
**Sentence 2:** A man in a blue snowsuit does a flip on a huge snowy mountain.
**Condition:** The person's connection with the ground.
**Gold Label:** 4 (Mostly equivalent)

---

**Few-shot Inference Prediction:** no(1)
**Reasoning:** *In Sentence 1... , the key action is jumping, which implies the person is airborne and not in contact with the ground. In Sentence 2... , the preposition 'on' suggest that he's on the mountain's surface. So, he's connected to the ground. Therefore, the two sentences have different connections with the ground. Hence, the answer is 'no(1)'.*

---

**SFT Baseline Prediction:** yes(3)

---

**PoLi-RL Prediction:** yes(4)
**Reasoning:** *Sentence 1 describes... . So he's airborne during the jump, meaning he's not connected to the ground. Sentence 2 talks about ... I infer that this action, similar to a jump or a trick in a snow-sports context, also means he is in the air. Therefore, they are similar because both have the person not connected to the ground. But the final score should be 4 because the action (jumping/flipping) leading to this state represent minor, differing details.*

Figure 4: A detailed case study comparing the reasoning processes and final outputs of the Few-shot, SFT baselines, and PoLi-RL on a C-STS sample that requires nuanced reasoning.

## 5 Conclusion

We introduce PoLi-RL, a two-stage reinforcement learning framework that resolves the complex credit assignment problem associated with listwise objectives in C-STS through a progressive curriculum and an innovative Parallel-Slice Ranking Reward (PSRR) mechanism. Our method establishes a new state-of-the-art for the cross-encoder architecture, significantly outperforming SFT baselines and even large models like GPT-4. As the first successful application of reinforcement learning to this task, our study validates a powerful paradigm for aligning LLMs with complex ranking objectives, demonstrating the framework's potential to other ranking-based tasks.

## REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. * sem 2013 shared task: Semantic textual similarity. In Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity, pp. 32–43, 2013.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70):1–53, 2024.

Ameet Deshpande, Carlos Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. C-STS: Conditional semantic textual similarity. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 5669–5690, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.345. URL https://aclanthology.org/2023.emnlp-main.345/.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. Transactions of the Association for Computational Linguistics, 8: 539–555, 2020.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Baixuan Li, Yunlong Fan, and Zhiqiang Gao. SEAVER: Attention reallocation for mitigating distractions in language models for conditional semantic textual similarity measurement. In Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 78–95, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.5. URL https://aclanthology.org/2024.findings-emnlp.5/.

Xianming Li and Jing Li. Bellm: Backward dependency enhanced large language model for sentence embeddings. arXiv preprint arXiv:2311.05296, 2023.

Rensis Likert. A technique for the measurement of attitudes. Archives of psychology, 1932.

Ziyong Lin, Quansen Wang, Zixia Jia, and Zilong Zheng. Varying sentence representations via condition-specified routers. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 17390–17401, 2024.

Xinyue Liu, Zeyang Qin, Zeyu Wang, Wenxin Liang, Linlin Zong, and Bo Xu. Conditional semantic textual similarity via conditional contrastive learning. In Proceedings of the 31st International Conference on Computational Linguistics, pp. 4548–4560, 2025.

Ayako Sato, Kyotaro Nakajima, Hwichan Kim, Zhousi Chen, and Mamoru Komachi. Tmu-hit's submission for the wmt24 quality estimation shared task: Is gpt-4 a good evaluator for machine translation? In Proceedings of the Ninth Conference on Machine Translation, pp. 529–534, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. arXiv preprint arXiv:2507.06229, 2025.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, et al. Ul2: Unifying language learning paradigms. arXiv preprint arXiv:2205.05131, 2022.

Jingxuan Tu, Keer Xu, Liulu Yue, Bingyang Ye, Kyeongmin Rim, and James Pustejovsky. Linguistically conditioned semantic textual similarity. arXiv preprint arXiv:2406.03673, 2024.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. arXiv preprint arXiv:2204.07705, 2022.

Kosuke Yamada and Peinan Zhang. Out-of-the-box conditional text embeddings from large language models. arXiv preprint arXiv:2504.16411, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

Young Hyun Yoo, Jii Cha, Changhyeon Kim, and Taeuk Kim. Hyper-cl: Conditioning sentence representations with hypernetworks. arXiv preprint arXiv:2403.09490, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

Jerrold H Zar. Spearman rank correlation. Encyclopedia of biostatistics, 7, 2005.

Bowen Zhang and Chunping Li. Pcc-tuning: Breaking the contrastive learning ceiling in semantic textual similarity. arXiv preprint arXiv:2406.09790, 2024a.

Bowen Zhang and Chunping Li. Advancing semantic textual similarity modeling: A regression framework with translated ReLU and smooth k2 loss. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 11882–11893, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.663. URL https://aclanthology.org/2024.emnlp-main.663/.

Gaifan Zhang, Yi Zhou, and Danushka Bollegala. Annotating training data for conditional semantic textual similarity measurement using large language models. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 27003–27015, 2025.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. Swift:a scalable lightweight infrastructure for fine-tuning, 2024. URL https://arxiv.org/abs/2408.05517.

# A Appendix

## A.1 Related Work

**Conditional Semantic Textual Similarity.** C-STS is a recent advancement over traditional STS that introduces a natural language condition to disambiguate the measurement of similarity between two texts. Research in this area has primarily established three mainstream paradigms: Bi-encoder, Tri-encoder, and Cross-encoder. Given two texts and a condition, the bi-encoder architecture typically uses a Siamese network to encode two text-condition pairs, while the tri-encoder architecture encodes each text and the condition separately before an aggregation step. A prevalent optimization strategy for these paradigms is contrastive learning. For instance, (Liu et al., 2025) propose a conditional contrastive learning framework that pulls representations of the same text pair under a high-similarity condition closer, while pushing them apart under a low-similarity one. Extending this, Hyper-CL (Yoo et al., 2024) utilizes a tri-encoder setup where a hypernetwork generates condition-specific projectors to dynamically adapt sentence representations within a contrastive objective. More recently, CSR introduced a parameter-free router for the tri-encoder, using the condition to re-weight sentence tokens to amplify relevant information without increasing model size.

In contrast, the cross-encoder architecture processes the concatenated texts and condition as a single input, enabling deep, token-level interaction. However, this theoretical advantage did not consistently translate to superior performance in earlier discriminative models. The state-of-the-art method in this setting, SEAVER, addressed this discrepancy by identifying that such models can be distracted by condition-irrelevant tokens. To resolve this, SEAVER (Li et al., 2024) introduces an attention reallocation mechanism that optimizes the model by re-weighting internal attention scores during fine-tuning, forcing a focus on the most salient information.

The advent of LLMs has introduced new approaches, mainly few-shot inference and using LLMs as feature extractors (Yamada & Zhang, 2025) (Li & Li, 2023). However, all these prior works, regardless of architecture, are confined to supervised paradigms like SFT or contrastive learning. Our work is the first to apply reinforcement learning to this task.

**Reinforcement Learning for Large Language Models.** Reinforcement Learning (RL) is pivotal for aligning and enhancing Large Language Models (LLMs), with algorithms like Proximal Policy Optimization (PPO) (Schulman et al., 2017) widely used to optimize non-differentiable objectives in tasks such as reasoning and code generation. To address PPO's limitations, such as high variance in long-sequence tasks, advanced variants have emerged.

Group Relative Policy Optimization (GRPO) (Guo et al., 2025) addresses these limitations by introducing a group-relative advantage estimation, which eliminates the need for a separate value function through Z-score normalization of rewards within sample groups. This approach simplifies training and enhances sample efficiency, as demonstrated in models like DeepSeek-R1. Building on this, Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025) provides an open-source, scalable RL system tailored for LLMs. DAPO incorporates key improvements, including dynamic sampling to adaptively adjust the number of generated completions based on reward variance, making it particularly effective for long-horizon reasoning tasks.

Leveraging DAPO's powerful optimization engine, our work, PoLi-RL, marks a significant departure by being the first to introduce a reinforcement learning framework to the C-STS task, thereby establishing a new optimization paradigm.

### A.2 Prompt Template for PoLi-RL

Below is the detailed few-shot prompt used for both the few-shot inference baseline and the training, evaluation process of PoLi-RL.

**Prompt for C-STS Task**

Judge the semantic similarity between Sentence 1 and Sentence 2 based **completely** on the given Condition. The final output must be exactly in this format: the similarity judgment ('yes' or 'no') followed by the score in parentheses, wrapped in <answer></answer> tags. Examples: <answer>yes(4)</answer>, <answer>no(1)</answer>. Include no other text, tags, or explanations.
To arrive at this output, follow these two steps:

**Step 1: Binary Judgment.** Determine if the sentences are 'similar' ('yes') or 'not similar' ('no').

- 'similar': The sentences are roughly, mostly, or completely equivalent under the condition.
- 'not similar': The sentences are dissimilar under the condition.

**Step 2: Fine-grained Score.** Assign an integer score based on Step 1:

- For a 'yes' judgment:
  * **5:** The two sentences are completely equivalent as they mean the same thing with respect to the condition.
  * **4:** The two sentences are mostly equivalent, but some unimportant details differ with respect to the condition.
  * **3:** The two sentences are roughly equivalent, but some important information differs or is missing with respect to the condition.
- For a 'no' judgment:

- **2:** The two sentences are dissimilar, but are on a similar topic with respect to the condition or shares a close semantic relationship. This applies when items are clearly different, but not direct opposites.
- **1:** The two sentences are dissimilar with respect to the condition, representing a direct opposition or a clear, unrelated difference. (e.g., 'man' vs. 'woman').

Here are some examples:

---

**Example 1:**
`<Sentence1>`: A girl is cooking in a kitchen and a man is standing next to her.
`<Sentence2>`: A man sitting with a pizza in his hand in front of pizza on the table.
`<Condition>`: The number of people.
`<answer>no(1)</answer>`
*Explanation: The first sentence mentions two people, while the second sentence mentions only one person.*

---

**Example 2:**
`<Sentence1>`: A wood table sitting by a wood framed bed with a lamp on it.
`<Sentence2>`: A microwave, refrigerator, television, and wooden drawers sit in the corner of a bedroom.
`<Condition>`: The room type.
`<answer>yes(5)</answer>`
*Explanation: We can infer from the two sentences that the room type are both bedroom.*

---

**Example 3:**
`<Sentence1>`: A small crowd gathered around the injured person.
`<Sentence2>`: A crowd jumps up and down to the tunes played by an artist.
`<Condition>`: The number of people
`<answer>yes(3)</answer>`
*Explanation: While both sentences mention crowds, it is important and unclear how many people there are.*

---

Now, apply these steps to the following sentences:

`<Sentence1>`: {sentence1}
`<Sentence2>`: {sentence2}
`<Condition>`: {condition}

### A.3 HYPERPARAMETER SETTINGS AND SENSITIVITY ANALYSIS

This section details the hyperparameter settings and provides empirical justification for selections not covered in the main paper. The default configurations used for our main results are summarized in Table 5.

Table 5: Default hyperparameter configurations for the main results.

| HyperParameter | Default Value |
|---|---|
| $\lambda_1 : \lambda_2 : \lambda_3$ | 1:0.25:1 |
| $\mu_1 : \mu_2 : \mu_3$ | 1:1.5:1 |
| $R_{\text{base}}$ | 0.5 |
| $G$ | 8 |
| $N$ | 24 |

In addition to the ablation studies presented in the main paper, we conducted sensitivity analyses on three key hyperparameters: the binary reward weights in Stage I, generation multiplicity ($G$) and the pairwise base reward ($R_{\text{base}}$).

**Sensitivity to Reward Weights in PoLi-RL Stage I.** In Stage I, the Binary Judgment Reward serves as an auxiliary signal to prevent the model from converging to ”safe” median scores. We test the sensitivity to the weight of this component ($\lambda_2$). Results in Table 6 show that PoLi-RL is highly robust to variations in $\lambda_2$, consistently establishing a strong foundation for Stage II training.

Table 6: Sensitivity to Binary Reward Weight ($\lambda_2$) in PoLi-RL's Stage I.

| **Method** | $\lambda_1$ **(Pointwise)** | $\lambda_2$ **(Binary)** | $\lambda_3$ **(Format)** | **Spearman** ↑ | **Pearson** ↑ |
|---|---|---|---|---|---|
| PoLi-RL (Stage I) | 1.0 | 0.1 | 0.1 | **44.94** | **44.93** |
| | 1.0 | 0.25 | 0.1 | 44.77 | 44.45 |
| | 1.0 | 0.5 | 0.1 | 44.67 | 44.42 |
| | 1.0 | 1.0 | 0.1 | 44.76 | 44.92 |

**Impact of Generation Multiplicity.** We investigate the trade-off between exploration diversity and computational overhead by varying $G \in \{4, 8, 12\}$. As shown in Table 7, while increasing $G$ theoretically aids RL exploration, empirical results indicate that $G = 4$ offers insufficient exploration coverage: while the restricted search space may suffice for basic scoring, it fails to generate the diverse candidates necessary for learning the fine-grained semantic distinctions in Stage II. Conversely, scaling to $G = 12$ leads to performance saturation and incurs higher computational overhead. These results validate our selection of $G = 8$ as the optimal configuration for balancing performance and efficiency.

Table 7: Ablation on generation multiplicity ($G$) in PoLi-RL both stages.

| $G$ | **Method** | **Spearman** ↑ | **Pearson** ↑ |
|---|---|---|---|
| 4 | PoLi-RL(Stage I) | 44.55 | 44.49 |
| 4 | PoLi-RL(Stage II) | 47.47 | 47.45 |
| 8 (original) | PoLi-RL(Stage I) | 44.77 | 44.45 |
| 8 (original) | PoLi-RL(Stage II) | **48.18** | **48.27** |
| 12 | PoLi-RL(Stage I) | 45.51 | 45.61 |
| 12 | PoLi-RL(Stage II) | 47.98 | 48.06 |

**Rationale for Pairwise Base Reward.** The pairwise reward (Eq. 7) incorporates a constant base reward $R_{base}$ to ensure a guaranteed positive signal when the model correctly predicts the ranking direction (ordinality), even if the exact score gap (cardinality) is imprecise. We compare the default $R_{base} = 0.5$ against removing it (0.0) or overweighting it (0.75). Table 8 demonstrates that 0.5 yields the best performance, justifying the need for a balanced reward structure.

Table 8: Sensitivity analysis of the pairwise base reward ($R_{\text{base}}$).

| **Method** | $R_{\textbf{base}}$ | **Spearman** ↑ | **Pearson** ↑ |
|---|---|---|---|
| PoLi-RL (Stage II) | 0 | 47.90 | 47.59 |
| | 0.25 | 47.78 | 47.81 |
| | 0.5 | **48.18** | **48.27** |
| | 0.75 | 47.54 | 47.33 |

### A.4 Performance across Different Model Sizes

To evaluate the efficiency of PoLi-RL and its performance across different parameter scales, we extend our experiments to include Qwen3-0.6B and Qwen3-4B backbones. Table 9 presents a comprehensive comparison against few-shot prompting, SFT baselines and previous SOTA methods.

The results provide compelling evidence for the efficacy of PoLi-RL. Our 0.6B model (44.34) achieves a massive 19.09 point improvement over its few-shot baseline and, remarkably, outperforms

Table 9: Performance comparison across different model sizes. We compare PoLi-RL against Few-shot, SFT baselines and previous SOTA methods across Qwen3-0.6B and 4B backbones.

| Methods | Training Paradigm | Parameters | Spearman ↑ | Pearson ↑ |
|---|---|---|---|---|
| *Previous SOTA on Discriminative Model and Generative Model* | | | | |
| SEAVER SimCSE$_{\text{LARGE}}$ | SFT | 355M | 43.83 | 43.81 |
| GPT-4 | Few-shot | - | 43.6 | - |
| *Our Implementation on Qwen3-0.6B* | | | | |
| Qwen3-0.6B | Few-shot | 0.6B | 25.25 | 25.19 |
| Qwen3-0.6B | SFT | 0.6B | 35.59 | 36.83 |
| PoLi-RL (Qwen3-0.6B) | RL | 0.6B | **44.34** | **44.36** |
| *Our Implementation on Qwen3-4B* | | | | |
| Qwen3-4B | Few-shot | 4B | 37.97 | 38.48 |
| Qwen3-4B | SFT | 4B | 38.41 | 39.45 |
| PoLi-RL (Qwen3-4B) | RL | 4B | **46.23** | **46.19** |
| *Our Implementation on Qwen3-8B* | | | | |
| Qwen3-8B | Few-shot | 8B | 37.9 | 38.54 |
| Qwen3-8B | SFT | 8B | 40.42 | 40.83 |
| PoLi-RL (Qwen3-8B) | RL | 8B | **48.18** | **48.27** |

both the proprietary giant GPT-4 (43.60) (Achiam et al., 2023) and the previous Cross-Encoder SOTA SEAVER (43.83) (Li et al., 2024). These findings demonstrate that the performance gains stem from PoLi-RL's ability to align reasoning processes with ranking objectives, rather than relying solely on large-scale parameters.

### A.5 Comparison with State-of-the-Art Reasoning Models

In this section, we benchmark PoLi-RL against current state-of-the-art general reasoning models. We evaluate GPT-4o and DeepSeek-R1 (Guo et al., 2025) on the C-STS test set using the same few-shot prompting setup described in Appendix A.2.

As shown in Table 10, our 8B model outperforms GPT-4o by 3.95 points and DeepSeek-R1 by 5.33 points. Even more remarkably, our 0.6B model slightly surpasses these massive proprietary models. This suggests that while proprietary models possess strong general reasoning capabilities, they struggle to strictly align with the fine-grained quantization standards of C-STS (i.e., the 1-5 Likert scale) in a few-shot setting. PoLi-RL bridges this gap by explicitly optimizing this alignment via RL, proving that a specialized, smaller model can surpass general-purpose giants on complex conditional ranking tasks.

Table 10: Benchmarking against state-of-the-art proprietary reasoning models under the same few-shot setting as PoLi-RL.

| Methods | Training Paradigm | Spearman ↑ | Pearson ↑ |
|---|---|---|---|
| DeepSeek-R1 | Few-shot | 42.85 | 42.36 |
| GPT-4o | Few-shot | 44.23 | 44.07 |
| PoLi-RL (Qwen3-0.6B) | RL | 44.34 | 44.36 |
| PoLi-RL (Qwen3-4B) | RL | 46.23 | 46.19 |
| PoLi-RL (Qwen3-8B) | RL | **48.18** | **48.27** |

### A.6 Comparison with Differentiable Ranking Objectives

In this section, we explicitly compare PoLi-RL against a strong cross-encoder regression baseline trained with differentiable ranking objectives. The goal is to determine whether optimizing a surrogate loss is sufficient to capture the rank-based nuances of C-STS. Specifically, we compare PoLi-RL against a baseline trained with the Pearson Correlation Coefficient (Pcc) Loss (Zhang & Li, 2024a), a state-of-the-art differentiable proxy for Spearman metric. Pcc loss is defined as:

$$\mathcal{L} = 1 - \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{10}$$

where $X$ represents the predicted scores and $Y$ is the ground-truth labels. Table 11 shows that while the regression baseline outperforms standard SFT, PoLi-RL still maintains a clear advantage (1.74). This superiority stems from the paradigm shift: Regression treats the model as a "black box" that maps embeddings to a scalar. In contrast, PoLi-RL optimizes the reasoning process itself. By aligning the token-level generation probability with the non-differentiable ranking metric, the model learns how to reason towards the correct score, enabling better generalization in complex conditional scenarios.

Table 11: Comparison between PoLi-RL and SFT with differentiable Pearson Correlation Coefficient (Pcc) Loss regression.

| Methods | Training Paradigm | Spearman ↑ | Pearson ↑ |
|---|---|---|---|
| Qwen3-8B | Few-shot | 37.90 | 38.54 |
| Qwen3-8B | SFT (Auto-regressive) | 40.42 | 40.83 |
| Qwen3-8B | SFT (Regression) | 46.44 | 46.59 |
| PoLi-RL (Qwen3-8B) | RL | **48.18** | **48.27** |

### A.7 Performance on Re-annotated C-STS Dataset

Recent work by Tu et al. (2024) identified potential label noise in the original C-STS dataset and released a re-annotated validation set. Subsequently, Zhang et al. (2025) further refined the dataset by correcting the condition descriptions and utilizing LLMs to clean the training labels. To ensure the robustness of our method against data quality issues, we re-evaluate PoLi-RL on the cleaner dataset provided by Zhang et al. (2025).

Table 12: Performance on the Re-annotated C-STS Dataset. Evaluation performed on the 30% hold-out split of the validation set following the protocol of Tu et al. (2024) and Zhang et al. (2025).

| Methods | Training Paradigm | Spearman ↑ | Pearson ↑ |
|---|---|---|---|
| Qwen3-8B | Few-shot | 64.42 | 64.50 |
| Qwen3-8B | SFT | 72.09 | 70.41 |
| PoLi-RL (Stage I) | RL | 74.74 | 73.49 |
| PoLi-RL (Stage II) | RL | **76.08** | **74.16** |

As detailed in Table 12, PoLi-RL maintains its significant performance advantage on the re-annotated data, achieving a 76.08 Spearman correlation and outperforming the SFT baseline by nearly 4 points. This confirms that our reported improvements are robust and valid, rather than an artifact of overfitting to label noise.

### A.8 Generalizability on Out-of-Domain Tasks

To empirically validate the generalizability of our framework beyond the C-STS domain, we applied PoLi-RL to the WMT-QE 2020 task (Fomicheva et al., 2020). This task shares the goal of optimizing Global Spearman correlation but differs fundamentally from C-STS in three aspects:

1. Different Domain: Multilingual translation quality estimation.
2. Different Scale: Continuous 0-100 scores.
3. Different Structure: Independent samples without adjacent pairing.

For this experiment, we simply disable the C-STS-specific Pairwise and Binary rewards, relying solely on Stage I (Pointwise) and Stage II (Listwise via PSRR). The prompt settings are adapted from Sato et al. (2024).

Table 13: Generalization performance on WMT-QE 2020 task (en-zh subset) using only the core PSRR mechanism, excluding C-STS-specific reward components.

| **Methods** | **Training Paradigm** | **Spearman ↑** | **Pearson ↑** |
|---|---|---|---|
| Qwen3-8B | Few-shot | 45.03 | 44.18 |
| Qwen3-8B | SFT | 50.90 | **51.09** |
| PoLi-RL (Stage I) | RL | 51.72 | 50.58 |
| PoLi-RL (Stage II) | RL | **54.33** | **51.09** |

As shown in Table 13, PoLi-RL achieves a 3.43 Spearman gain over the strong SFT baseline. This empirically validates the adaptability of the PSRR mechanism to ranking tasks with fundamentally different label scales and data structures.

### A.9 LLM USAGE STATEMENT

The large language model (LLM) was utilized during the preparation of this manuscript. The use of this technology was strictly confined to the role of a writing assistant for the sole purpose of improving the linguistic quality of the text. Specifically, the LLM was employed for tasks related to grammar, syntax, phrasing, and overall readability. Its function was exclusively to perform surface-level linguistic refinements on text already written by the human authors. Crucially, the LLM did not contribute to any substantive or intellectual aspects of the research. The conceptualization of the study, the design of the methodology, the execution of experiments, the interpretation of results, and the formulation of conclusions were all executed by the human authors.