RL from Teacher-Model Refinement: Gradual Imitation Learning for Machine Translation

Anonymous ACL submission

Abstract

Preference-learning methods for machine translation (MT), such as Direct Preference Optimization (DPO), have shown strong gains but typically rely on carefully curated preference triplets and often struggle to generalize beyond their tuning domains. We propose *Reinforcement Learning from Teacher-Model Refinement (RLfR)*, which replaces static triplets with *on-policy, actor-conditioned refinements* produced by a frozen teacher. At each step, the actor samples candidate translations, the teacher performs a minimal local edit of each draft, and the actor is reinforced to close the gap using a composite reward that combines *scaled negative edit distance* for lexical and structural fidelity with *COMET* for semantic adequacy. This formulation yields a stable, model-aware learning signal without requiring curated preference triplets, making it naturally scalable to many language pairs without additional data collection. Experiments on FLORES-200 (*en↔de/es/zh/ko/ja*) show that RLfR consistently outperforms strong baselines, improving semantic quality and entity preservation across automatic, LLM-judge, and human evaluations.

1 Introduction

Recent breakthroughs in large language models (LLMs) (Achiam et al., 2023; Grattafiori et al., 2024; Team, 2024; Yang et al., 2025) have markedly improved multilingual Machine Translation, providing strong zero-shot and few-shot capabilities and motivating work on adapting general-purpose LLMs to production-grade MT (Hoang et al., 2023; Xu et al., 2023; Alves et al., 2024; Xu et al., 2024a).

Machine Translation (MT) has advanced through supervised fine-tuning (SFT) and preference-learning methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023; Agrawal et al., 2024; Tang et al., 2025; Li et al., 2024; Ramos et al., 2024). Although effective, these methods often require curated triplet datasets built from anchor translations and ranked alternatives, making data construction costly, slow to iterate, and difficult to scale across languages and domains.

Recent work (Donato et al., 2022; Guo et al., 2025) has explored off-policy reinforcement learning as an alternative to supervised fine-tuning, with strong results in structured domains such as mathematics and code generation. Extending these methods to machine translation, however, faces two key obstacles. First, translation admits multiple valid outputs, which makes robust rule-based rewards difficult to design and brittle under lexical and syntactic variation. Second, off-policy updates trained against static references or previously sampled responses often provide limited returns for their computational cost: fixed targets can drift from the model's own predictions and frequently converge near the initial checkpoint (Table 11).

We address both issues by delegating credit assignment to a frozen teacher (e.g., GPT-4o-mini) that produces context-sensitive refinements for each actor output. Conditioning feedback on the model's own translations yields minimal, targeted edits that preserve sentence structure while steadily narrowing the actor–teacher gap, resulting in a more effective and model-aware reinforcement learning signal.

We present RL from Teacher-model Refinement (RLfR), a reinforcement-learning framework that eliminates triplet construction. For every source sentence, a frozen teacher model (GPT-4o-mini) refines the actor's draft translation $\hat{y}$ into a higher-quality target y^*. Lexical and semantic gaps between $\hat{y}$ and y^* form the *learning signal* and are quantified by two complementary metrics: (i) edit distance $\left(R_{\text{edit}}\right)$

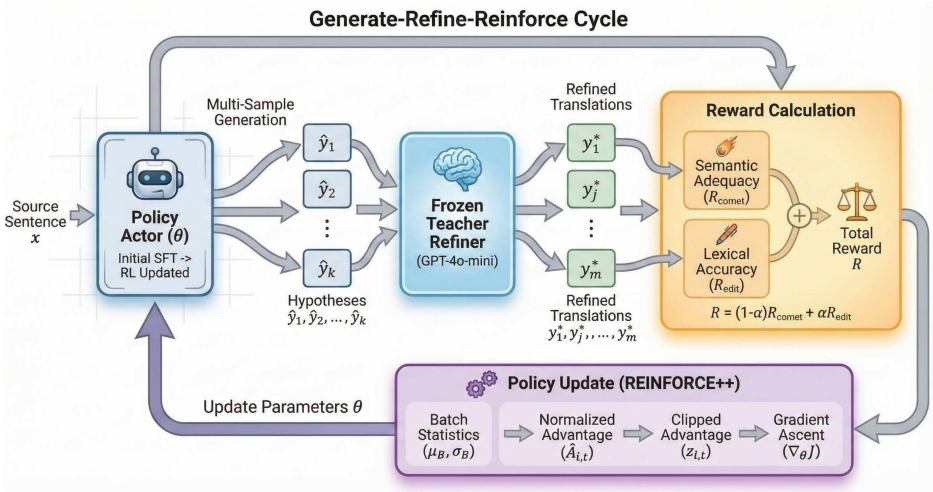


Figure 1: RLfR follows an incremental generate–refine–reinforce cycle. A policy actor is first initialised, then refined online by a frozen teacher model, and finally updated with a REINFORCE++ objective that uses multi-sampling.

and (ii) COMET (R_{comet}). These scores are combined into a single reward, and a critic-free REINFORCE++ (Hu, 2025) update maximises the expected composite reward, continually pushing the actor toward the refiner. For evaluation only, we additionally report the M-ETA (Conia et al., 2024) metric to assess entity-level fidelity. Our contribution can be summarized as follows:

- *Online refinement learning:* RLfR replaces static triplet data with *actor-conditioned* online refinements from a frozen refiner, removing dataset curation and frozen references.
- *Progressive, multi-metric learning:* A single reward signal that blends edit-distance and COMET shrinks the actor–refiner gap at every step, delivering simultaneous gains in lexical fidelity, semantic adequacy, and entity correctness.
- *Strong empirical gains:* On the FLORES-200 benchmark (en↔de/es/zh/ko/ja) RLfR improves COMET, reduces lexical errors, and lowers M-ETA entity mistakes, outperforming diverse baselines.

These properties make RLfR a robust, scalable, and data-efficient solution for multilingual MT.

2 RL from Teacher-Model Refinement (RLfR)

RLfR follows an incremental generate–refine–reinforce cycle (Fig. 1). A supervised actor is first initialised, then repeatedly refined online by a frozen teacher model, and finally updated with a critic-free REINFORCE++ objective that uses multi-sampling.

2.1 Supervised Actor Initialisation

Although pre-trained or initially supervised fine-tuned (SFT) multilingual models offer broad coverage, their performance on machine translation (MT) benchmarks frequently remains insufficient for high-quality evaluation settings. Prior studies (Guo et al., 2025) further indicate that reinforcement learning (RL) benefits substantially from being initialized with a well-trained SFT model, rather than from a weaker or zero-shot baseline.

However, publicly available parallel corpora vary substantially in quality, domain coverage, and consistency across language pairs. To obtain a stronger and more reliable starting point for RL, we leverage GPT-4o-mini to distill a high-quality multilingual parallel corpus. This synthetic supervision provides consistent quality while remaining significantly more cost-efficient than human-curated data or large-scale

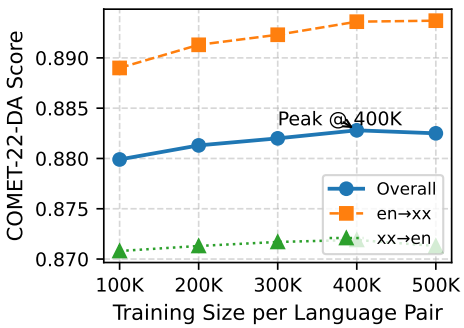


Figure 2: COMET scores on the FLORES test set using the LLaMA-3.1 8B model, evaluated with different amounts of distilled SFT examples.

preference annotations (see Appendix B).

To assess how far static distillation alone can strengthen the base model, we fine-tune our actor using GPT-4o-mini–translated parallel datasets derived from the FineFineWeb corpus (Zhang et al., 2024). We construct identical samples across five language pairs and vary the amount of training data from 100K to 500K sentence pairs per language pair. This controlled setup allows us to directly measure the effect of increased offline teacher supervision on translation quality.

The actor parameters θ are optimized from the initial checkpoints by minimizing the standard cross-entropy objective:

$$\mathcal{L}_{\text{SFT}} = \frac{1}{N} \sum_{i=1}^{B} \sum_{t=1}^{T_i} -\log P(y_{i,t} \mid x_i, y_{i,<t}), \quad (1)$$

where N denotes the number of predicting tokens in a mini batch.

As shown in Figure 2, translation quality improves as more distilled data is used, but gains plateau beyond approximately 400K examples. Importantly, the 400K setting achieves the best overall COMET score (0.8828), whereas larger distillation sets yield no further gains despite more supervision. While the marginal cost of synthetic pairs is low—and our online refinement is similarly inexpensive (Appendix B)—the plateau suggests that the limitation of offline distillation is not cost but the static nature of its supervision. This, in turn, motivates more targeted, model-aware refinement signals as introduced in our RLfR framework. Consequently, we adopt the 400K SFT model as the initialization for all subsequent RLfR experiments.

2.2 Online Target Refinement via Multi-Sample Generation

Given a source sentence x, we first sample multiple translation hypotheses from the actor model:

$$\hat{y}_{1:k} \sim p_\theta(y \mid x) \quad (2)$$

Each hypothesis $\hat{y}_i$ is concatenated with the original source prompt x and then passed to a frozen teacher refinement model $\mathcal{T}$ (GPT-4o-mini), yielding refined target translations:

$$y_i^* = \mathcal{T}(x, \hat{y}_i) \quad (3)$$

Unlike offline distillation, this refinement is *actor-conditioned*: the teacher observes the model's hypothesis and produces a minimal, locally improved target rather than a static translation. This provides supervision that adapts to the actor's current behaviour and cannot be replicated by simply scaling distilled data.

By generating multiple hypotheses per input, the actor obtains a richer view of the local reward landscape, significantly reducing variance during policy gradient updates once batch-normalised advantages are computed.

2.3 Reward Design and Policy Updates

Reward Formulation We design the reward function to balance semantic adequacy and lexical accuracy:

$$R(\hat{y}, y^*) = (1-\alpha) \underbrace{R_{\text{comet}}}_{\text{Semantic Adequacy}} + \alpha \underbrace{R_{\text{edit}}}_{\text{Lexical Accuracy}}, \quad \alpha \in [0,1] \quad (4)$$

Here, R_{edit} is the scaled negative Levenshtein distance between the hypothesis $\hat{y}$ and the refined target y^*:

$$R_{\text{edit}} = s\left(1 - \frac{\text{EditDist}(\hat{y}, y^*)}{\max\{|\hat{y}|, |y^*|\}}\right), \quad (5)$$

where the scaling function $s(z)$ discretizes lexical similarity scores based on dynamic empirical quantiles calculated from a subset of the training corpus (`rollout_batch_size`):

$$s(z) = \begin{cases} -1 & z < \text{mean}, \\ \frac{z-\text{mean}}{Q_{90}-\text{mean}} & \text{mean} \le z < Q_{90}, \\ 1 & z \ge Q_{90}, \end{cases} \quad (6)$$

Here, mean and Q_{90} denote the average and the 90th percentile, respectively, dynamically computed from the subset samples. We use Q90 as a robust high-similarity anchor; in preliminary runs, nearby choices such as Q80 and Q95 produced similar trends.

The semantic reward R_{comet} is the COMET (Rei et al., 2022) score linearly mapped to $[-1, 1]$. We explore three reward combinations: lexical-focused ($\alpha = 1$), semantic-focused ($\alpha = 0$), and mixed ($\alpha = 0.5$).

Batch-normalised REINFORCE++ Objective We now provide a detailed description of the RLfR training procedure, as summarized in Algorithm 1. Let B denote the minibatch of sampled hypotheses obtained from a single rollout, with each sample $i \in B$ associated with a scalar reward R_i. The advantage of each token t in sequence i of length T_i is defined as:

$$A_{i,t} = R_i - \beta \sum_{j=t}^{T_i} \mathrm{KL}\Big(\pi_\theta(\cdot|\hat{y}_{i,<j}, x_i) \,\|\, \pi_{\theta_{\text{old}}}(\cdot|\hat{y}_{i,<j}, x_i)\Big)$$

where the Kullback-Leibler (KL) divergence term penalizes deviation from the reference policy $\pi_{\theta_{old}}$ for all tokens from position t to the end of the sequence, and β controls the strength of this regularization.

To stabilize the reward signal and mitigate variance during policy gradient estimation, we introduce a batch normalization strategy applied to the advantage. The batch-level advantage statistics are computed as follows:

$$\mu_B = \frac{1}{N} \sum_{i \in B} \sum_{t=1}^{T_i} A_{i,t}, \tag{7}$$

$$\sigma_B = \sqrt{\frac{1}{N} \sum_{i \in B} \sum_{t=1}^{T_i} (A_{i,t} - \mu_B)^2 + \varepsilon}, \tag{8}$$

where μ_B represents the mean advantage across the batch, σ_B is the standard deviation, $N = \sum_{i \in B} T_i$ is the total number of tokens in the batch, and ε is a small constant (e.g., 10^{-6}) included for numerical stability.

Utilizing these batch-level statistics, we compute a batch-normalized advantage for each sample i at position t:

$$\hat{A}_{i,t} = \frac{A_{i,t} - \mu_B}{\sigma_B}. \tag{9}$$

This normalized advantage metric is subsequently employed to scale gradient updates on a per-sample basis, thereby stabilizing the learning process.

To prevent excessively large policy updates that might destabilize training, we further incorporate a clipped importance-weighted advantage inspired by Proximal Policy Optimization (PPO):

$$\begin{aligned} \rho_{i,t} &= \frac{\pi_\theta(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t})}{\pi_{\theta_{old}}(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t})}, \\ z_{i,t} &= \mathrm{clip}\big(\rho_{i,t},\, 1-\epsilon,\, 1+\epsilon\big)\, \hat{A}_{i,t} \end{aligned} \tag{10}$$

where $\pi_\theta(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t})$ denotes the probability of selecting token $\hat{y}_{i,t}$ at position t under the current policy parameterized by θ, conditioned on the input x_i and all previous tokens in the sequence, and $\pi_{\theta_{old}}(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t})$ represents the corresponding probability under the reference policy—a snapshot of parameters prior to the most recent update. The clipping operation constrains the importance weight to the interval $[1-\epsilon, 1+\epsilon]$, thereby limiting the magnitude of individual policy updates at each token position and improving training stability.

Gradient ascent is then performed using the batch-normalized and clipped policy gradient estimation across all tokens in the batch, defined as:

$$\nabla_\theta J(\theta) = \frac{1}{N} \sum_{i \in B} \sum_{t=1}^{T_i} z_{i,t} \cdot \nabla_\theta \log \pi_\theta(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t}). \tag{11}$$

This formulation of the objective significantly enhances training stability and convergence properties, effectively balancing exploration and exploitation while ensuring robust policy updates.

3 Experiments

3.1 Experimental Setup

Supervised MT-SFT Baselines. We fine-tune several pre-trained LLMs—such as LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen3-1.7B (Yang et al., 2025), and ZLM-2.3B (Zoom, 2025)—on distilled parallel data

Baseline & Method	COMET (%) — XX→EN						COMET (%) — EN→XX					
	de	es	ja	ko	zh	**Avg.**	de	es	ja	ko	zh	**Avg.**
LLaMA-3.1-8B as Baseline												
Baseline	88.72	86.29	87.18	87.28	86.46	87.19	88.73	87.10	91.73	90.37	88.85	89.36
Semantic-Focused	89.64	87.55	88.44	88.36	87.60	**88.32**	88.87	87.41	91.90	90.54	89.00	**89.54**
Lexical-Focused	89.50	87.34	88.12	88.19	87.27	88.08	88.79	87.34	91.74	90.38	88.89	89.43
Mix	89.55	87.47	88.28	88.29	87.51	88.23	88.82	87.38	91.81	90.46	88.80	89.45
Qwen3-1.7B as Baseline												
Baseline	88.50	86.01	87.07	86.76	86.40	86.95	87.71	86.41	91.02	89.68	88.53	88.67
Semantic-Focused	88.96	86.49	87.62	87.46	86.98	87.50	88.04	86.86	91.30	90.18	88.73	**89.02**
Lexical-Focused	88.85	86.38	87.62	87.45	86.94	87.45	87.95	86.62	91.26	90.05	88.58	88.89
Mix	89.02	86.47	87.69	87.58	87.07	**87.57**	87.92	86.66	91.24	90.06	88.69	88.91
ZLM-2.3B as Baseline												
Baseline	88.36	86.02	86.86	87.01	85.97	86.84	88.06	86.53	91.48	89.97	88.08	88.82
Semantic-Focused	89.11	86.95	87.69	87.96	86.78	**87.70**	88.03	86.88	91.72	90.27	88.48	**89.08**
Lexical-Focused	88.98	86.79	87.57	87.74	86.62	87.54	88.06	86.74	91.57	90.01	88.29	88.93
Mix	89.09	86.93	87.68	88.00	86.77	87.69	88.17	86.81	91.63	90.11	88.38	89.02

Table 1: COMET (%) scores for both XX→EN and EN→XX directions across three SFT baselines. Bolded values indicate best average scores in each block. **Semantic-Focused**, **Lexical-Focused**, and **Mix** correspond to RLfR variants that adjust the composite reward weighting between COMET and edit-distance.

comprising 400K sentence pairs for each translation direction (en → xx, xx → en). Each model is optimized via cross-entropy loss to learn a strong supervised policy $p_\theta(y \mid x)$.

Off-line Preference Learning. Starting from each MT-SFT baseline, we generate two candidate translations for every source prompt: one produced by the MT-SFT model and one supplied by GPT-4o-mini as a distilled reference. We then apply reference-free COMET to score each $\langle \text{prompt}, \text{candidate} \rangle$ pair. The candidate with the higher COMET score is labeled as "positive" and the other as "negative." These static triplets are subsequently used to train the model via Direct Preference Optimization (DPO), isolating the gains achieved purely through preference learning.

RL with Fixed References. From each MT-SFT checkpoint, we sample 5K examples per direction from the same distilled corpus and treat the GPT-4o-mini outputs as immutable ground truth. Actors are updated using the standard REINFORCE++ objective against these fixed rewards to quantify the benefits of RL when the teacher's outputs remain static.

RL from Refinement (RLfR, proposed). Using each MT-SFT checkpoint, we subsample 5K prompts per direction and draw k hypotheses from the actor. Each hypothesis is sent to teacher for refinement, and actors are trained with batch-normalised REINFORCE++ loss using dynamic rewards. This configuration evaluates the efficacy of continuous, high-quality feedback from an external teacher model.

All experiments—supervised, off-line preference learning, and reinforcement learning—use subsets of the same distilled dataset to ensure a controlled and directly comparable evaluation. Teacher outputs are generated using GPT-4o-mini with a fixed low-temperature decoding setup; full configuration details are provided in Appendix A. We additionally report standalone teacher calibration results in Appendix C.

3.2 Experimental Results

RLfR is Effective in Improving Semantic Precision and Fluency. We evaluate semantic precision and fluency improvements from RLfR using COMET across two translation directions (XX→EN and EN→XX) and multiple model scales: a large model (LLaMA-3.1-8B) and smaller models (Qwen3-1.7B, ZLM-2.3B).

Table 1 shows that RLfR consistently improves COMET for both directions across all baselines. For LLaMA-3.1-8B, the *Semantic-Focused* variant achieves the best average COMET in both directions, outperforming the Lexical-Focused and Mixed configurations, indicating stronger semantic adequacy and fluency. The same trend holds for smaller models, where Semantic-Focused RLfR provides steady gains over the baselines, demonstrating robustness across model sizes.

RLfR Effectively Preserves Named En-

Baseline & Method	M-ETA (%) per Language						Gain
	de	es	ja	ko	zh	**Avg.**	
LLaMA-3.1-8B as Baseline							
Baseline	28.45	40.60	26.42	32.89	4.43	26.56	—
Mix	33.65	45.47	28.08	37.05	4.29	29.71	**(+3.15)**
Lexical-Focused	33.11	46.68	27.11	36.78	4.43	29.62	(+3.06)
Semantic-Focused	32.01	44.93	27.52	37.58	4.72	29.35	(+2.79)
Qwen3-1.7B as Baseline							
Baseline	23.94	33.83	21.02	25.91	4.85	21.91	—
Mix	24.49	33.69	22.41	29.4	4.99	23.00	**(+1.09)**
Lexical-Focused	24.35	35.18	22.41	28.05	4.57	22.91	(+1.00)
Semantic-Focused	23.53	33.83	22.54	26.31	4.71	22.18	(+0.27)
ZLM-2.3B as Baseline							
Baseline	21.20	35.45	20.47	27.25	3.32	21.54	—
Mix	22.57	38.02	23.65	31.81	3.74	23.96	**(+2.42)**
Lexical-Focused	21.61	36.94	23.24	31.95	3.74	23.50	(+1.96)
Semantic-Focused	22.85	37.21	21.16	31.68	4.02	23.38	(+1.84)

Table 2: M-ETA (%) performance across five languages under three SFT baselines. Each block reports baseline and enhanced methods. Best-performing variants are bolded.

Prompt	How does Jia Yingchun contribute to the overall narrative of the Dream of the Red Chamber?
Reference	贾迎春对红楼梦的整体叙述有何贡献？ 贾 (Jia: family name), 迎 (welcome), 春 (spring)
SFT Output	贾莹春如何为《红楼梦》的整体叙述做出贡献？ 莹 (crystal or glittering): semantically incorrect
RLfR Output	贾英春如何为《红楼梦》的整体叙述做出贡献？ 英 (heroic or outstanding): phonetically accurate, semantically appropriate

Figure 3: Comparison of translations for "Jia Yingchun." SFT introduces semantic distortion, whereas RLfR produces a phonetically faithful and contextually valid variant.

tities Across Languages. Accurate named-entity translation is crucial for factual consistency and user trust, especially in high-stakes domains. We evaluate entity preservation with M-ETA across a large model and smaller models in five target languages.

Table 2 shows that RLfR consistently improves M-ETA across model scales. Gains are strongest for the large model, and the *mixed* configuration (semantic-focused + lexical-focused) achieves the best overall performance, suggesting that jointly optimizing both signals yields superior entity preservation. Smaller models also benefit, with smaller but consistent improvements; the mixed strategy remains best, followed by semantic- or lexical-focused variants depending on the baseline.

We observe smaller measured gains in Chinese, which we attribute primarily to limitations of reference-based metrics rather than reduced RLfR effectiveness. Chinese references often contain inconsistent or localized renderings of named entities, so reference-overlap metrics can penalize valid variants. As illustrated in Figure 3, the reference uses the canonical literary form 贾迎春, while RLfR produces 贾英春, a phonetically faithful and semantically reasonable modern variant; SFT outputs 贾莹春, introducing a semantic shift. Because M-ETA rewards strict agreement with the reference, it can underestimate real-world improvements in Chinese where orthographic variation is common. Overall, RLfR strengthens named-entity preservation across languages and model sizes, with the mixed objective delivering the most reliable gains.

4 Analysis

4.1 Effectiveness over Baseline Approaches

We compare RLfR against two baselines—DPO and RL with Fixed References—using an ablation study on LLaMA-3.1-8B. This controlled setup isolates the effect of actor-conditioned refinement as a post-training strategy under matched data, model, and reward settings.

As shown in Table 3: (i) **RLfR achieves the best COMET and M-ETA**, improving both semantic adequacy and entity fidelity

Method	COMET (Avg.)	M-ETA (Avg.)
MT-SFT (baseline)	88.23	26.56
DPO	88.63	27.19
RL w/ Fixed Ref.	88.80	28.20
RLfR (Ours)	**88.93**	**29.35**

Table 3: Comparison of average COMET and M-ETA scores using LLaMA-3.1-8B. Best-performing results are bolded.

via model-aware, iterative teacher refinement. Importantly, M-ETA is used only as an evaluation metric and is not part of the RLfR reward, indicating that the entity-fidelity gains are not directly optimized by the training reward; (ii) **RL with Fixed References improves over DPO** but remains below RLfR, likely because static, fully rewritten targets provide a weaker improvement signal relative to the model's own outputs (see Appendix Table 11); and (iii) **DPO shows the smallest gains**, suggesting that preference-only supervision offers limited corrective guidance for fine-grained translation control.

To further test whether RLfR's semantic gains extend beyond COMET-specific effects, we evaluate LLaMA-3.1-8B variants using BLEURT-20 and MetricX-24, neither of which is used as a training reward. This allows us to separately assess RLfR-Lexical, which does not use COMET, and RLfR-Mix/RLfR-COMET, which do. As shown in Table 4, all RLfR variants outperform SFT and DPO on both metrics. RLfR-Lexical improves despite using no COMET reward, while RLfR-Mix and RLfR-COMET also generalize to metrics outside the reward. Together with the M-ETA gains in Table 3, these results indicate that RLfR improves semantic adequacy and entity fidelity beyond direct metric optimization.

Method	BLEURT-20	MetricX-24
SFT	0.315	2.62
DPO	0.330	2.51
RLfR-Lexical	0.344	2.41
RLfR-Mix	0.346	2.40
RLfR-COMET	**0.348**	**2.37**

Table 4: Decoupled semantic evaluation on LLaMA-3.1-8B using BLEURT-20 and MetricX-24. BLEURT is higher-is-better; MetricX-24 is lower-is-better.

Method	Sem.	Entity	Fluency	Halluc. (↑)
SFT	4.358	4.476	4.300	4.632
DPO	4.534	4.572	4.375	4.705
RL w/ Fixed Ref.	4.500	4.580	4.460	4.740
RLfR (Ours)	**4.570**	**4.636**	**4.548**	**4.836**

Table 5: Fine-grained LLM-as-a-judge scores over 500 examples. Scores were assigned by `gpt-5-mini`, which is independent from the `gpt-4o-mini` teacher used in RLfR training.

Comparison	RLfR Win	Tie	Baseline Win
RLfR vs. SFT	231	126	143
RLfR vs. DPO	200	121	179
RLfR vs. Fixed Ref.	215	140	145

Table 6: Pairwise preferences from 500 judged comparisons per baseline. RLfR is consistently preferred over all alternatives.

4.2 LLM-as-a-Judge Evaluation

To complement automatic metrics, we evaluate LLaMA-3.1-8B outputs with `gpt-5-mini`, which is independent from the RLfR teacher `gpt-4o-mini`. The judge scores 500 examples, evenly split between En→X and X→En across five languages, using both fine-grained criteria and pairwise preferences. As shown in Tables 5 and 6, RLfR achieves the highest scores for semantic adequacy, entity fidelity, fluency, and hallucination control, and is preferred over all baselines.

To reduce judge-family bias, we further run `gemini-2.5-flash-lite`, a non-OpenAI judge, on 1,000 pairwise comparisons per baseline, again evenly split between En→X and X→En across the same five languages.

Table 7 confirms the same trend: RLfR wins 75.8%, 73.9%, and 72.8% of non-tied comparisons against SFT, DPO, and Fixed-Reference RL, respectively. Together, the two judges suggest that RLfR's gains are robust to evaluator choice rather than tied to a single model family.

4.3 Human Evaluation on Hard Translation Subsets

To validate RLfR beyond automatic metrics and LLM judges, we conduct blinded human pairwise evaluations on hard FLORES-200 subsets for en↔ko and en↔zh, with 50 examples per language pair. We construct the subsets by filtering for sentences that are entity-dense, containing proper nouns, numbers, dates, or units, or structurally complex, involving multi-clause

Comparison	RLfR Win	Tie	Baseline Win	Win Rate
RLfR vs. SFT	499	342	159	75.8%
RLfR vs. DPO	473	360	167	73.9%
RLfR vs. Fixed Ref.	456	373	171	72.8%

Table 7: Pairwise preferences from 1,000 `gemini-2.5-flash-lite`-judged comparisons per baseline. Win rates are computed excluding ties.

Comparison	RLfR Win	Tie	Baseline Win	Win Rate
RLfR vs. SFT	63	22	15	80.8%
RLfR vs. DPO	57	26	17	77.0%

Table 8: Human pairwise preferences on hard subsets from en↔ko and en↔zh, with 50 examples per language pair. Win rates are computed excluding ties.

constructions, coordination, enumeration, or punctuation-heavy phrasing. We then briefly verify the selected examples manually. Annotators compare RLfR against SFT and DPO, prioritizing adequacy and factual/entity fidelity, using fluency as a tie-breaker, and assigning multi-select reason tags for *Entity/Factual fidelity* and *Fluency*.

Table 8 shows that human annotators prefer RLfR over SFT and DPO, with non-tie win rates of 80.8% and 77.0%. Table 9 further attributes RLfR wins mainly to entity/factual fidelity, with additional fluency gains. These results confirm that RLfR improves difficult cases requiring factual precision and structural adequacy.

Qualitative analysis. We provide qualitative comparisons and success/failure cases in Appendix D, illustrating the effectiveness of actor-conditioned refinement.

5 Related Work

LLM-based Machine Translation Large language models (LLMs) are increasingly used for machine translation, with recent work improving multilingual capacity and translation quality through language expansion (Yang et al., 2023; Cui et al., 2025), reasoning-enhanced translation (Liu et al., 2025), fine-tuning and knowledge distillation (Xu et al., 2023; Li et al., 2024; Stap et al., 2024), plug-and-play modules and adaptive rejection (Xu et al., 2024a), gradient-based data selection (Pan et al., 2024), scaling-law analysis (Isik et al., 2025), and document-level translation with large-scale corpora (Pal et al., 2024).

Reason Tag	SFT	DPO	RLfR
Entity/fidelity-driven wins	27%	35%	**68%**
Fluency-driven wins	40%	45%	**55%**

Table 9: Reason-tag attribution for non-tie human preferences. Percentages are computed over non-tie comparisons where the corresponding system wins. Tags are multi-select and therefore do not sum to 100%.

Preference Learning in Machine Translation Preference-based MT improves supervised fine-tuning with human feedback, quality estimation, or fine-grained automatic criteria. Prior work has used PPO with quality estimation models (Schulman et al., 2017; He et al., 2024), span-level xCOMET rewards for fine-grained reward modeling (Guerreiro et al., 2024; Ramos et al., 2024), DPO and PPO-style extensions (Rafailov et al., 2023; Zhong et al., 2024), and automatic metrics such as XCOMET to construct preference datasets (Agrawal et al., 2024). Xu et al. (2024b) proposed Contrastive Preference Optimization (CPO) with COMET-based reward selection (He et al., 2024), while Li et al. (2024) used LLM-generated feedback and analogy data for targeted distillation. However, most approaches still rely on curated triplets or fixed references, which can be disconnected from the model's current predictions.

6 Conclusion

We introduced RLfR, which replaces static preference triplets with actor-conditioned teacher refinements. Across five FLORES-200 language pairs, RLfR improves lexical fidelity and semantic adequacy, outperforming MT-SFT, DPO, and fixed-reference RL in COMET and M-ETA across model scales. Decoupled metrics, LLM-judge evaluations, and human preferences further show that these gains extend beyond direct metric optimization.

Limitations

Teacher Bias Because RLfR depends on teacher-generated translations, any biases or systematic errors in the teacher model can be directly inherited by the student. This makes the strength and calibration of the teacher particularly important for preventing the amplification of undesirable behaviors. We mitigate evaluation bias via independent judging, but training-time teacher bias remains a limitation.

Teacher Invocation Cost Although RLfR requires multiple teacher queries during refinement, the actual cost depends on how the teacher is accessed. When using API-based models, monetary cost can be a concern, whereas locally deployable strong models reduce API cost and allow batching or hardware-level optimizations. Nonetheless, large-scale refinement still demands careful resource–performance trade-offs.

References

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

Sweta Agrawal, José GC de Souza, Ricardo Rei, António Farinhas, Gonçalo Faria, Patrick Fernandes, Nuno M Guerreiro, and Andre Martins. 2024. Modeling user preferences with automatic metrics: Creating a high-quality preference dataset for machine translation. *arXiv preprint arXiv:2410.07779*.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Simone Conia, Daniel Lee, Min Li, Umar Farooq Minhas, Saloni Potdar, and Yunyao Li. 2024. Towards cross-cultural machine translation with retrieval-augmented generation from multilingual knowledge graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16343–16360, Miami, Florida, USA. Association for Computational Linguistics.

Menglong Cui, Pengzhi Gao, Wei Liu, Jian Luan, and Bin Wang. 2025. Multilingual machine translation with open large language models at practical scale: An empirical study. *arXiv preprint arXiv:2502.02481*.

Domenic Donato, Lei Yu, Wang Ling, and Chris Dyer. 2022. Mad for robust reinforcement learning in machine translation. *arXiv preprint arXiv:2207.08583*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

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

Zhiwei He, Xing Wang, Wenxiang Jiao, Zhuosheng Zhang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2024. Improving machine translation with human feedback: An exploration of quality estimation as a reward model. *arXiv preprint arXiv:2401.12873*.

Hieu Hoang, Huda Khayrallah, and Marcin Junczys-Dowmunt. 2023. On-the-fly fusion of large language models and machine translation. *arXiv preprint arXiv:2311.08306*.

Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.

Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and Sanmi Koyejo. 2025. Scaling laws for downstream task performance in machine translation. In *The Thirteenth International Conference on Learning Representations*.

Jiahuan Li, Shanbo Cheng, Shujian Huang, and Jiajun Chen. 2024. Mt-patcher: Selective and extendable knowledge distillation from large language models for machine translation. *arXiv preprint arXiv:2403.09522*.

Sinuo Liu, Chenyang Lyu, Minghao Wu, Longyue Wang, Weihua Luo, Kaifu Zhang, and Zifu Shang. 2025. New trends for modern machine translation with large reasoning models. *arXiv preprint arXiv:2503.10351*.

Proyag Pal, Alexandra Birch-Mayne, and Kenneth Heafield. 2024. Document-level machine translation with large-scale public parallel corpora. In *The 62nd Annual Meeting of the Association for Computational Linguistics*, pages 13185–13197. Association for Computational Linguistics (ACL).

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. 2024. G-dig: Towards gradient-based diverse and high-quality instruction data selection for machine translation. *arXiv preprint arXiv:2405.12915*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Miguel Moura Ramos, Tomás Almeida, Daniel Vareta, Filipe Azevedo, Sweta Agrawal, Patrick Fernandes, and André FT Martins. 2024. Fine-grained reward optimization for machine translation using error severity mappings. *arXiv preprint arXiv:2411.05986*.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

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

David Stap, Eva Hasler, Bill Byrne, Christof Monz, and Ke Tran. 2024. The fine-tuning paradox: Boosting translation quality without sacrificing llm abilities. *arXiv preprint arXiv:2405.20089*.

Zilu Tang, Rajen Chatterjee, and Sarthak Garg. 2025. Mitigating hallucinated translations in large language models with hallucination-focused preference optimization. *arXiv preprint arXiv:2501.17295*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2412.15115*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*.

Haoran Xu, Kenton Murray, Philipp Koehn, Hieu Hoang, Akiko Eriguchi, and Huda Khayrallah. 2024a. X-alma: Plug & play modules and adaptive rejection for quality translation at scale. *arXiv preprint arXiv:2410.03115*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024b. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

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

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023. Bigtranslate: Augmenting large language models with multilingual translation capability over 100 languages. *arXiv preprint arXiv:2305.18098*.

Ge Zhang, Xinrun Du, Zhimiao Yu, Zili Wang, Zekun Wang, Shuyue Guo, Tianyu Zheng, Kang Zhu, Jerry Liu, Shawn Yue, Binbin Liu, Zhongyuan Peng, Yifan Yao, Jack Yang, Ziming Li, Bingni Zhang, Minghao Liu, Tianyu Liu, Yang Gao, Wenhu Chen, Xiaohuan Zhou, Qian Liu, Taifeng Wang, and Wenhao Huang. 2024. Finefineweb: A comprehensive study on fine-grained domain web corpus.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2024. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*.

Zoom. 2025. How we're preparing for the next era of ai. Accessed: 2025-06-28.

A Implementation Details

Teacher Generation Settings All teacher outputs used for data distillation, preference learning, and refinement are generated using `gpt-4o-mini` under a fixed low-temperature, deterministic-like decoding setup (temperature = 0.1, top-p = 1.0, max_tokens = 2048). We use the default chat-completions API interface, and all prompts follow the system–user format shown below to ensure consistency across all stages of training.

Prompt used for refinement:

You are an expert in Machine Translation and Cultural Localization. Refine a given machine-translated sentence by improving its quality and naturally localizing named entities into the target language only.
Guidelines

- **Entity Localization**: Translate or adapt all named entities into natural, target-language-only forms. Do not include the source language. Use official or culturally standard localized names.
- **Style and Tone**: Match the tone of the source (formal, neutral, conversational).
- **Fluency**: Ensure the result is natural, idiomatic, and fluid.
- **Grammar and Accuracy**: Fix any awkward grammar or inconsistent wording.

B Teacher Cost for `gpt-4o-mini`

We estimate the monetary cost of the refinement stage, which is the only component of RLfR requiring repeated teacher queries. On average, each teacher input contains 63 tokens when averaged across the five target languages (de, es, ja, ko, zh) and both en → xx and xx → en directions. The RL dataset consists of 5K prompts per direction, but the actual number of `gpt-4o-mini` calls is determined by the rollout configuration. At each RL step we sample `rollout_batch_size` = 560 prompts and draw $n_{\text{samples}} = 8$ hypotheses per prompt, all of which are refined by the teacher. For S RL steps this yields

$$N_{\text{calls}} = S \times 560 \times 8,$$

which corresponds to $N_{\text{calls}} \approx 1{,}792{,}000$ teacher invocations when $S = 400$.

Assuming the `gpt-4o-mini` pricing scheme with per-million prices $p_{\text{in}} = \$0.15$ for input tokens and $p_{\text{out}} = \$0.60$ for output tokens, we define the per-token costs $c_{\text{in}} = p_{\text{in}}/10^6$ and $c_{\text{out}} = p_{\text{out}}/10^6$. Denoting the average refinement length by L_{out}, the total cost of the refinement stage is

$$\text{Cost}_{\text{refine}} = N_{\text{calls}}\big(63\, c_{\text{in}} + L_{\text{out}}\, c_{\text{out}}\big).$$

For translation, the output length is typically comparable to the input length across the five language pairs, so we approximate $L_{\text{out}} \approx 63$. This yields roughly 1.1×10^8 input tokens and 1.1×10^8 output tokens, corresponding to a total cost of about \$80–90 for a single RLfR run with $S = 400$ steps. The same formula applies to any alternative teacher, with only the pricing parameters changing.

C Alternative Teacher Models and Plug-and-Play Refinement

RLfR treats the teacher as a black-box refiner: the algorithm only assumes access to a model that can improve or rewrite candidate translations, and does not depend on a specific architecture or family. To assess robustness to the teacher choice, we run small-scale experiments with alternative refiners for the LLaMA-3.1-8B baseline, including full `gpt-4o` and `gpt-5` in addition to our default `gpt-4o-mini` teacher.

To keep teacher cost manageable, we restrict these ablations to the Korean and Chinese directions. For each teacher, we fine-tune RLfR with the same hyperparameters and report COMET scores obtained by averaging XX→EN and EN→XX for each language (ko, zh).

All RLfR settings improve over the SFT baseline. As shown in Table 10, `gpt-4o-mini` and `gpt-4o` are nearly indistinguishable in COMET, while `gpt-5` provides only slightly higher scores at a higher inference cost. In practice, `gpt-4o-mini` offers a substantially more cost-efficient option with only marginal degradation in refinement quality on these benchmarks.

Teacher	COMET ↑ ko | zh
SFT Baseline (no RLfR)	88.83 | 87.66
RLfR (`gpt-4o-mini`)	89.38 | 88.16
RLfR (`gpt-4o`)	89.39 | 88.16
RLfR (`gpt-5`)	89.42 | 88.18

Table 10: COMET scores for RLfR with alternative teacher models on LLaMA-3.1-8B. We evaluate only Korean (ko) and Chinese (zh) to control teacher cost. For each language, we report a single COMET score obtained by averaging the XX→EN and EN→XX directions; each cell shows ko | zh.

We note that `gpt-5` is currently our strongest general-purpose teacher model. However, this work was finalized in late May 2025, before the public release of `gpt-5`. As a result, we did not re-run all main experiments with `gpt-5` as the default refiner and instead include it only in this small-scale ablation. A more comprehensive comparison with `gpt-5` teachers is left for future work.

Overall, these results indicate that RLfR is *teacher-agnostic*: it can flexibly adopt higher-capacity, more economical, or domain-specialized refinement models, as long as they provide reasonably strong machine translation quality.

Standalone teacher calibration. We also evaluate the default refinement teacher, `gpt-4o-mini`, as a standalone translator on the same FLORES-200 evaluation set using the same inference prompt and decoding constraints as RLfR inference. It achieves average COMET scores of 88.39 for X→En and 89.46 for En→X across five languages, and an average M-ETA score of 30.06. These results serve as a calibration reference rather than a strict ceiling. For example, in Table 1, the LLaMA-3.1-8B RLfR student with the semantic-focused reward reaches 89.54 COMET on En→X, slightly above the teacher's 89.46 under the same evaluation. This is expected because RLfR uses the teacher to refine student drafts rather than to provide static translations from scratch, making the supervision actor-conditioned and targeted to student-specific errors.

D Qualitative Analysis: Supervision Comparison, Successes, and Failures

Input (Korean)
넷플릭스 시리즈 '오징어 게임' 시즌3의 제작발표회가 9일 오전 서울 용산구 드래곤시티에서 열렸다. 시즌3에서는 다시 게임에 참여하게 된 기훈, 신분을 숨기고 침투하는 프런트맨, 그리고 살아남은 참가자들의 마지막 운명이 그려진다.

SFT Model Output (Draft)
The third season of the Netflix series 'Ongiye Game' was officially launched at Seoul Dragon City on September 9 in the morning. The Ozone Game Season 3 tells the story of 'Gintetsu', a game participant who loses his closest friend after rejoining the game with a new purpose, 'Frontman', who hides his identity and enters the game, and the final fate of the participants who survive in this brutal game.

Comment: *Contains entity/name errors and awkward phrasing (e.g., Ongiye Game, Gintetsu, Ozone Game).*

RLfR-Refined Output
The third season of the Netflix series Squid Game was officially unveiled at Seoul Dragon City on the morning of September 9. The new season follows Gi-hun, who reenters the deadly game with a renewed purpose but ends up losing his closest friend; the mysterious Front Man, who conceals his identity while infiltrating the game; and the final fate of the remaining participants who survive its brutal trials.

Comment: *Corrects key entities (e.g., Squid Game, Gi-hun) and improves fluency while preserving the original structure.*

Fixed Reference from Teacher
On the morning of the 9th, a production presentation for Netflix's Squid Game Season 3 was held at Seoul Dragon City in Yongsan-gu, Seoul. Squid Game Season 3 follows the story of Gi-hun, who reenters the game with a personal mission but ends up losing his closest friend; the Front Man, who hides his identity while infiltrating the game; and the final destinies of the survivors of the brutal competition.

Table 11: Comparison of outputs under different supervision styles. The SFT Model Output contains name errors and awkward phrases, while the RLfR-refined output corrects key entities and improves fluency—providing a more effective and model-aware reward signal.

This appendix provides qualitative comparisons and multilingual success/failure cases for RLfR. First, Table 11 compares a model prediction, a fixed teacher reference, and an RLfR-refined output for the same Korean input. The example illustrates a limitation of fixed references:

Entity-Level Success (en→kr)

Source: In its decision, the PUB said it had concerns about bill affordability for Manitoba consumers, recommending the province development a program to assist lower-income Hydro customers.

SFT: 판결에서 PUB는 매니토바 소비자들의 청구서 affordability에 우려를 표명하며, 저소득 하이드로 고객을 지원하는 프로그램을 개발할 것을 권고했습니다.

RLfR: PUB는 판결에서 **매니토바 소비자들의 요금 부담에 대한 우려를 표명**하며, **저소득층 하이드로 고객을 지원하는 프로그램을 개발할 것**을 권고했습니다.

Comment: RLfR *correctly rephrases* `"bill affordability"` as `"요금 부담"` and *preserves* named entities like PUB, Manitoba, and Hydro — showing stronger fidelity.

Semantic-Level Success (kr→en)

Source: 젊은 음악가였던 시절, 토마스는 이 지역에서 잘 알려진 로이 헤드와 트레잇츠의 로이 헤드와 친구가 되어 함께 활동했습니다.

SFT: While he was a young musician, Thomas befriended Roy Head of Roy Head and the Traits, who was well known in the area.

RLfR: During his time as a young musician, Thomas **became friends with Roy Head from the well-known local band, Roy Head and the Traits**, and they **worked together**.

Comment: RLfR *retains and refines* the relationship and sequence of events, adding clarity while preserving meaning.

Entity-Level Failure (en→kr)

Source: 12x12" oil on canvas panel.

RLfR: 30x30 cm **캔버스 패널에 유화.**

Comment: RLfR *converts imperial to metric* (`"12x12 in"`→`"30x30 cm"`), *altering source meaning* — a fidelity loss.

Semantic-Level Failure (kr→en)

Source: 구리.

RLfR: Guri.

Comment: RLfR *mistranslates* the chemical element `copper` as the Korean city **"Guri"** based on phonetics — a clear semantic error.

Table 12: Korean (`kr`) examples showcasing RLfR's strengths (blue) and weaknesses (red) in entity fidelity and semantic accuracy.

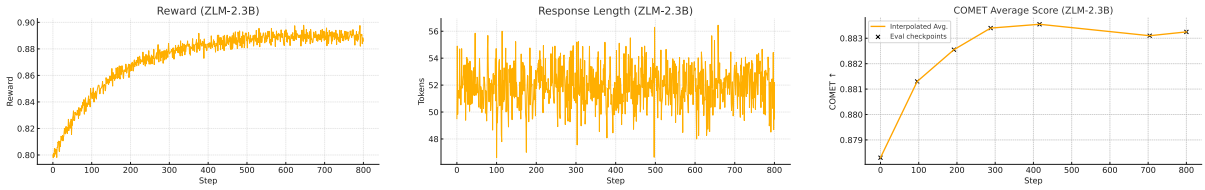
they are often complete rewrites that diverge from the model's original phrasing and structure. Although such references can be fluent and correct, their dissimilarity provides a weaker reward signal about what the actor should change. In contrast, RLfR conditions the teacher on the model output and the source query, producing minimal, targeted edits that preserve structure while correcting factual and stylistic errors.

We further include multilingual success and failure cases in Table 12. These examples show that RLfR can improve entity preservation and semantic adequacy, while also revealing remaining fidelity errors such as unintended unit conversion or incorrect disambiguation.

E Stepwise Analysis of Teacher Mimicry

We track teacher-assigned reward, response length, and COMET during RLfR training in Figure 4.

Reward increases steadily, indicating closer alignment with the teacher's refinements, while response length remains tightly bounded, suggesting that gains are not driven by length manipulation. COMET peaks around step 400 and remains stable afterward, indicating durable improvements without reward overfitting or regression.



(a) Reward trajectory (b) Response-length stability (c) COMET-22 (avg)

Figure 4: Stepwise dynamics of teacher-guided refinement for the ZLM-2.3B model trained with a mixture of supervised and refinement-based updates. Reward rises monotonically, response length stays within a narrow band, and COMET-22 peaks at step 416—aligning with the reward plateau—before marginally tapering off.

F RLfR Training Algorithm

Algorithm 1 RLfR: Reinforcement Learning from Refinement

Require: Mini-batch of inputs $\mathcal{B}$; actor policy π_θ with parameters θ; frozen teacher-refiner $\mathcal{T}$; number of samples k; learning rate η

Ensure: Updated actor parameters θ

1: **for** $x \in \mathcal{B}$ **do** ▷ Process each input in the batch

2: **Sample** k **hypotheses** from the current actor:

$$\hat{y}_{1:k} \sim \pi_\theta(\cdot \mid x)$$

3: **Obtain refined targets** from the teacher:

$$y^*_{1:k} \leftarrow \mathcal{T}(x, \hat{y}_{1:k})$$

4: **Compute per-sample rewards**:

$$R_{1:k} \leftarrow \textsc{ComputeReward}(\hat{y}_{1:k},\, y^*_{1:k})$$

5: **end for**

6: **Compute batch-level statistics:**

$$(\mu_B, \sigma_B) \leftarrow \textsc{BatchStats}(R)$$

7: **Normalize advantages:**

$$A_{1:k} \leftarrow \frac{R_{1:k} - \mu_B}{\sigma_B}$$

8: **Update actor parameters using REINFORCE++:**

$$\theta \leftarrow \theta + \eta \, \nabla_\theta \mathcal{L}_{\text{R++}}(\theta; A_{1:k})$$

Algorithm 1 summarizes RLfR as a *generate–refine–reinforce* cycle. For each input, the actor samples k hypotheses, the frozen teacher refines them, and rewards are computed between drafts and refinements. Batch-normalized advantages are then used in a critic-free REINFORCE++ update.