PAMT: Process-Aligned Reinforcement Learning for Multi-Domain Machine Translation

Anonymous ACL submission

Abstract

Multi-domain machine translation (MDMT) requires more than fluent generation: it demands domain-sensitive translation decisions such as domain disambiguation, terminology control, and stylistic adaptation. Large reasoning models (LRMs) make such decisions explicit through intermediate translation steps, but our analysis across 15 domains and four translation directions shows that this explicit reasoning is double-edged: it improves long-form and high-difficulty translation, yet often drifts in terminology-intensive and stylistically constrained settings. We trace this failure to a credit-assignment bottleneck: existing methods optimize final outputs or coarse trajectories, but cannot identify which translation steps actually help the final translation. To address this, we propose PAMT, a process-aligned training framework that combines cold-start domain-aware Long-CoT supervision with reinforcement learning. PAMT uses sequence-level format and outcome rewards for the final translation, together with a step-level process reward that measures how much each explicit translation step increases the likelihood of the reference translation. Experiments on two backbones show that PAMT consistently outperforms MT-specialized baselines and remains competitive with strong LLMs/LRMs across in-domain, OOD, and multilingual settings.

1 Introduction

Multi-domain machine translation (MDMT) requires more than semantic adequacy. To translate faithfully across domains, a model must make domain-sensitive decisions about ambiguity resolution, terminology, and style (Saunders, 2022; Jiang et al., 2020; Lai et al., 2022; Zheng et al., 2024a; Hu et al., 2024; Man et al., 2025). However, most translation methods based on large language models (LLMs) still treat translation as a direct sequence generation task (Xu et al., 2024a; Guo et al., 2024; Pang et al., 2025), without explicitly modeling the intermediate reasoning process. This makes reasoning-oriented translation particularly appealing: explicit intermediate translation steps can, in principle, expose the decision process needed for domain-faithful translation (He et al., 2024). Yet it remains unclear when these decisions help MDMT and when they fail.

To investigate this question, we first conduct a systematic comparison of LLMs and LRMs across 15 domains and four translation directions (Section 2). Our analysis reveals a clear split. Explicit reasoning is most helpful on long-context and high-difficulty inputs (Section 2.1), where translation benefits from stepwise decomposition and refinement. However, it is much less reliable in terminology-intensive and stylistically constrained settings (Section 2.2), where seemingly plausible reasoning can drift away from domain-specific conventions. These results highlight the need to supervise and control the translation decisions within the translation process.

Recent work has gradually moved from merely exposing the translation process to optimizing it. Workflow-based methods first make the process explicit, but only through a few coarse stages (Chen et al., 2024a; Feng et al., 2025c; Wang et al., 2024b). Chain-of-Thought (CoT)-based methods then provide more detailed process traces (Hu et al., 2024; He et al., 2024; Wang et al., 2025a), yet still learn them mainly through offline imitation. Reinforcement learning (RL)-based reasoning-augmented methods further optimize translation with reward signals (He et al., 2025; Feng et al., 2025a; Wang et al., 2025c,d; Yang et al., 2025b; Li et al., 2025), but these signals still operate largely on final outputs or whole trajectories. As a result, supervision remains misaligned with the underlying translation decisions. This raises a central question: How can intermediate translation decisions become creditable and

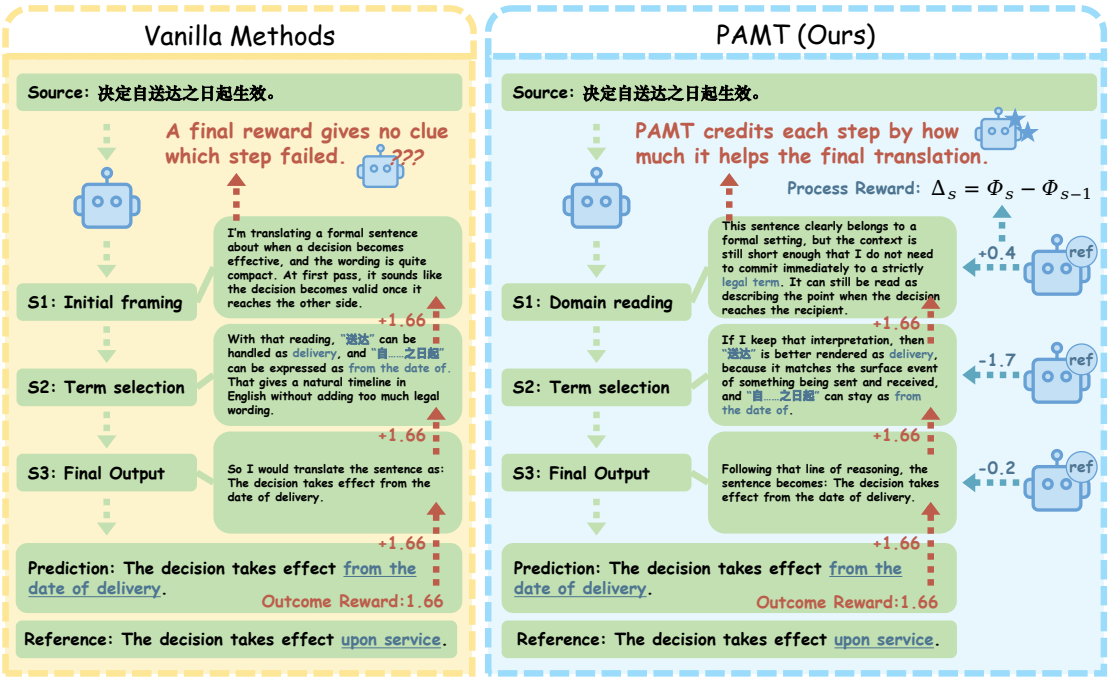

Figure 1: Overview of vanilla reasoning-augmented MT and PAMT. PAMT aligns both the translation process and the final output, addressing the misaligned credit assignment of vanilla methods.

optimizable for domain-faithful translation?

We address this problem with **PAMT**, a **P**rocess-**A**ligned training framework for MD**MT**. PAMT first performs cold-start supervised fine-tuning (SFT) on distilled domain-aware Long-CoT data to initialize an explicit translation process. It then applies RL to align the translation process with the final translation outcome. Specifically, final translation quality is optimized with sequence-level format and outcome rewards, while the translation process is optimized with a step-level reward defined by how much each explicit translation step increases the likelihood of the reference translation. This process-output alignment breaks the credit-assignment bottleneck in reasoning-augmented MT, allowing the model to optimize not only final translation quality which translation decisions support domain-faithful translation.

Our contributions are threefold. **(1)** We conduct a systematic evaluation of LLMs and LRMs for MDMT across 15 domains and four translation directions, showing that explicit translation reasoning helps in long-form and high-difficulty settings but often fails in terminology-intensive and stylistically constrained ones. **(2)** We identify a key bottleneck in reasoning-augmented MT: supervision granularity is misaligned with decision granularity, making intermediate translation steps difficult to credit and optimize, thereby leading to terminology and style drift. **(3)** We propose PAMT, a process-aligned training framework that mitigates this bottleneck by aligning translation processes with final translation outcomes, and show on two backbones that it is competitive with SOTA LLMs/LRMs while outperforming MT-specialized baselines across in-domain, out-of-domain (OOD), and multilingual settings.

Model	Document-Level			Sentence-Level		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI
GPT-4o	39.62	87.39	73.79	36.61	89.13	84.14
DeepSeek-V3	**40.43**	87.28	73.50	**36.63**	89.17	84.15
Gemini-2.0-Flash	39.55	87.00	73.09	36.62	**89.20**	83.99
OpenAI-o1	35.62	**87.78**	**76.19**	35.26	88.20	**85.63**
DeepSeek-R1	32.22	86.62	74.12	35.80	88.66	84.09
OpenAI-o3-mini	35.57	87.35	74.01	35.38	88.96	83.91
Gemini-2.0-Flash-Thinking	40.15	87.33	73.42	36.56	89.09	83.90

Table 1: Performance at the document and sentence levels.

2 Preliminary Evaluation

Setup. We compare LLMs and LRMs across 15 domains and four translation directions (En↔Zh, En↔De), analyzing explicit translation processes along four dimensions: input length, difficulty, MQM-style quality, and terminology accuracy. Details are given in Appendix B.

2.1 When Explicit Processes Help

Document-level Evaluation. To isolate the effect of longer context, we compare sentence-level and

Model	Level 1			Level 2			Level 3			Level 4			Level 5		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI
GPT-4o	**41.96**	**85.13**	79.80	30.08	84.88	82.15	26.49	82.75	80.91	22.61	83.11	75.48	24.94	83.59	67.11
DeepSeek-V3	40.81	85.09	79.90	27.90	84.86	82.20	25.43	82.54	80.97	22.57	82.95	75.30	25.06	83.30	67.64
Gemini-2.0-Flash	41.15	84.39	79.46	29.22	84.93	81.96	27.53	82.80	80.68	24.53	82.72	74.57	28.93	83.91	66.28
OpenAI-o1	40.34	84.59	**81.10**	29.67	84.88	**83.28**	25.76	**82.94**	**81.94**	21.99	**83.31**	**76.92**	22.78	**84.02**	**69.62**
DeepSeek-R1	41.43	84.33	79.86	27.12	84.25	81.75	24.32	82.15	80.44	18.66	82.17	75.27	19.53	82.43	68.78
OpenAI-o3-mini	40.74	84.12	79.44	26.49	84.56	82.14	24.93	82.36	80.73	21.09	82.67	75.37	22.70	83.27	67.63
Gemini-2.0-Flash-Thinking	39.87	83.84	79.36	**30.78**	**84.98**	82.21	**28.15**	82.68	80.62	**24.54**	83.02	74.71	**28.97**	83.93	66.32

Table 2: Performance by translation complexity for LRMs and traditional LLMs.

document-level translation on WMT22 (Table 1). BLEU increases for most models when moving to document-level inputs, indicating that longer context often helps lexical matching and local consistency. By contrast, COMET and COMETKiwi drop almost universally, suggesting that document-level translation mainly stresses semantic adequacy and contextual reliability. More importantly, this degradation is consistently smaller for LRMs than for LLMs. For example, the drop for GPT-4o is larger than for OpenAI-o1 on both COMET (1.74 vs. 0.42) and COMETKiwi (10.35 vs. 9.44). Overall, LLMs retain the strongest BLEU, while LRMs achieve the strongest COMETKiwi and become more competitive on COMET at the document level. These results suggest that as translation moves from isolated sentences to full documents, the bottleneck shifts from local generation to process-level decision making.

Impact of Translation Difficulty. To examine how translation difficulty affects model behavior, we use DeepSeek-V3 to assign source sentences from Multi-Domain (De→En), WMT22 (De↔En and Zh↔En), and Guofeng WebNovel (Zh→En) into five difficulty levels (Figure 4; Table 2). A clear pattern emerges: while traditional LLMs perform competitively on easy inputs, their performance drops sharply as difficulty increases, whereas LRMs surpass them from Level 2 onward and maintain consistent advantages on COMET and COMETKiwi. This gap indicates that performance at higher difficulty is limited not by local generation, but by the translation process itself. As inputs become more complex, success increasingly depends on process-level decisions such as ambiguity resolution, compositional decomposition, and consistency-preserving revision—capabilities that are better supported by explicit reasoning.

2.2 When Explicit Processes Hurt

MQM Error Analysis. We adopt the MQM hierarchy and use DeepSeek-V3 as an automatic scoring agent following Freitag et al. (2021), focusing on fine-grained error categories rather than aggregate scores. As shown in Table 3, LRMs tend to incur slightly fewer Accuracy and Fluency errors, but noticeably more style and terminology errors, especially in domains with stronger stylistic and lexical constraints such as law and medical. This pattern is important: the dominant failure of LRMs is not semantic breakdown, but process drift. In many cases, the model reaches a translation that remains globally plausible while deviating from the domain-faithful path in register, stylistic convention, or lexical choice. This explains why stronger reasoning alone is insufficient for MDMT. Once translation quality depends on fine-grained domain constraints, final-output supervision becomes too coarse: it can judge whether the sentence is good overall, but cannot tell which intermediate translation decisions preserved or violated style and terminology. The MQM results therefore highlight the need to supervise the translation process itself, rather than only its final outcome.

Model	Acc. (%) ↓	Fluency (%) ↓	Non-trans (%) ↓
GPT-4o	50.78	7.03	0.23
DeepSeek-V3	50.72	6.84	0.09
Gemini-2.0-Flash	52.86	6.87	0.16
OpenAI-o1	50.95	7.10	0.15
DeepSeek-R1	50.50	6.68	**0.08**
OpenAI-o3-mini	**49.97**	7.19	0.12
Gemini-2.0-Flash-Thinking	52.54	**6.38**	0.15
Model	**Style (%) ↓**	**Term. (%) ↓**	**Term Acc. (%) ↑**
GPT-4o	30.58	11.38	37.67
DeepSeek-V3	30.51	11.83	37.92
Gemini-2.0-Flash	**29.43**	**10.68**	**38.55**
OpenAI-o1	29.58	12.22	37.42
DeepSeek-R1	30.83	11.91	37.83
OpenAI-o3-mini	30.48	12.24	37.38
Gemini-2.0-Flash-Thinking	29.64	11.29	37.62

Table 3: MQM (↓) and terminology accuracy (↑) results for various LLMs and LRMs on De→En. The top section reports MQM error rates for accuracy (Acc.), fluency, and non-translation (Non-trans), while the bottom section reports style, terminology (Term.), and terminology accuracy (Term. Acc.).

Terminological Accuracy. We further isolate term-level behavior using the WMT23 Terminology

Shared Task (Semenov et al., 2023), which provides bilingual sentence pairs with aligned source–target terms. For each example, we provide only the source sentence and measure whether the expected target term is correctly produced in the translation (Table 3). The results reinforce the MQM findings: terminology accuracy is substantially less stable than the gains observed on general semantic metrics, and LRMs underperform in settings where translation depends more on faithful lexical realization. This shows that stronger reasoning does not automatically yield stronger term control. Instead, terminology translation is a process-level decision that must remain aligned with domain constraints throughout generation. Together with the MQM analysis, these results point to the same conclusion: the main bottleneck of LRMs in MDMT lies in how intermediate translation steps are supervised.

3 Related Work

Prior work makes the translation process explicit at different granularities to improve translation quality. Workflow-based methods (He et al., 2024; Briakou et al., 2024; Chen et al., 2024b,a; Wang et al., 2024b; Ki and Carpuat, 2024; Feng et al., 2025c) decompose translation into fixed stages such as drafting, evaluation, and refinement. Later work introduces domain-aware CoT (Hu et al., 2024) or multi-agent trajectories (Wang et al., 2025b) for SFT, but still learns the process mainly by imitating traces constructed offline. This further motivates RL-based reasoning-augmented MT, where the translation process and final output are optimized through outcome-level feedback from translation metrics (Feng et al., 2025a; He et al., 2025) or exemplar-enhanced rewards (Wang et al., 2025d). DeepTrans (Wang et al., 2026) and TAT-R1 (Li et al., 2025) further introduce process- and output-level rewards through external LLM scoring and terminology constraints. Yet these rewards are still assigned over the whole process, making it difficult to identify which intermediate translation step causes an unfaithful translation output.

4 Method

To address the mismatch between explicit translation steps and their optimization, we propose PAMT, a two-stage process-aligned framework consisting of a cold-start stage for explicit process initialization and an RL stage for joint process-output alignment.

4.1 Cold Start

The goal of this stage is to activate the base model's reasoning ability and ensure that it can produce an explicit translation process. We therefore begin with a cold-start SFT stage using a Long-CoT translation dataset distilled from DeepSeek-R1. The dataset contains approximately 7k translation reasoning examples across 10 diverse domains, with around 700 examples per domain. Each example follows the same structured format, `<think>...</think><answer>...</answer>`, where the model first generates the translation process and then outputs the final translation.

4.2 RL Stage

Task Setup. During RL, each training example consists of a source sentence x_i and a reference translation y_i^*. Conditioned on x_i, the policy π_θ generates a structured response

$$o_i = \texttt{<think>}z_i\texttt{</think><answer>}y_i\texttt{</answer>}, \quad (1)$$

where z_i denotes the explicit translation process and y_i is the final translation. To obtain step-level supervision, we split the `<think>` span by double newlines and represent the resulting process as a sequence of K_i reasoning steps: $z_i = [z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(K_i)}]$.

Reward design. We use both sequence-level and step-level rewards. The sequence-level rewards preserve supervision on the final response, while the step-level reward explicitly evaluates the usefulness of intermediate reasoning steps.

For response format, we use a simple binary reward: $r_i^{\text{fmt}} = 1$ if $\text{Valid}(o_i) = 1$, and $r_i^{\text{fmt}} = -1$ otherwise, where $\text{Valid}(\cdot)$ checks whether the output follows the structured format defined in Section 4.1. For translation quality, we define the outcome reward as:

$$r_i^{\text{out}} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \widetilde{q}_i, \quad (2)$$

where $\mathcal{Q} = \{\text{BLEU}, \text{COMET}, \text{COMETKiwi}\}$. These two rewards assess whether the final response is well-formed and of high quality, but they do not tell the model which reasoning steps are responsible for that quality.

To provide process-level supervision, we score each reasoning prefix using a frozen reference model π_{ref}. For the prefix containing the first k reasoning steps, let $z_{i,\leq k}$ denote their concatenation, and define the corresponding context as

$$c_{i,k} = [x_i, \texttt{<think>}, z_{i,\leq k}, \texttt{</think><answer>}]. \quad (3)$$

We then define the *process potential* of this prefix as the teacher-forced log-likelihood of the reference translation y_i^* under π_{ref}:

$$\phi_{i,k} = \sum_{m=1}^{|y_i^*|} \log \pi_{\text{ref}}\big(y_{i,m}^* \mid c_{i,k}, y_{i,<m}^*\big)\,. \quad (4)$$

Intuitively, $\phi_{i,k}$ measures how supportive the current reasoning prefix is for producing the reference translation. If adding a new reasoning step makes the reference translation easier to predict, that step is likely helpful; if it makes the reference less predictable, that step is likely noisy or misleading. Based on this idea, we define the *process gain* of step k as the improvement in potential after adding that step:

$$r_{i,k}^{\text{proc}} = \phi_{i,k} - \phi_{i,k-1}, \qquad k = 1, \ldots, K_i, \quad (5)$$

where $\phi_{i,0}$ is computed from the empty reasoning prefix. Thus, $r_{i,k}^{\text{proc}}$ captures the marginal contribution of step k to making the reference translation more predictable.

Because policy optimization is performed at the token level, we further distribute each step gain uniformly over the tokens inside that step. Specifically, for a token t belonging to step $z_i^{(k)}$, we assign

$$r_{i,t}^{\text{proc}} = \frac{r_{i,k}^{\text{proc}}}{|z_i^{(k)}|}. \quad (6)$$

This preserves the total gain of each step while avoiding a bias that would otherwise favor longer steps simply because they contain more tokens.

Credit Assignment. We next combine sequence-level and process-level signals into a single token reward sequence. Let t_i^{last} denote the position of the last valid response token. The token reward is defined as

$$r_{i,t} = \mathbf{1}[t = t_i^{\text{last}}]\big(r_i^{\text{fmt}} + r_i^{\text{out}}\big) + \lambda\, r_{i,t}^{\text{proc}}, \quad (7)$$

where λ controls the weight of process reward. This design cleanly separates the roles of the two signals. The sequence-level rewards are placed at the end of the response, reflecting overall format correctness and translation quality, while the process reward provides dense feedback on reasoning tokens. From the token reward sequence, we compute the return-to-go at each position:

$$R_{i,t} = \sum_{u=t}^{T_i} r_{i,u}, \qquad R_i^{\text{traj}} = R_{i,1}. \quad (8)$$

Here $R_{i,t}$ represents the cumulative future reward from token t onward, and R_i^{traj} is the total return of the trajectory.

Optimization. For the G rollouts sampled from the same source sentence, GRPO normalizes token-level returns using group-level trajectory statistics to obtain the advantage at each position:

$$A_{i,t} = \frac{R_{i,t} - \mu_g}{\sigma_g + \epsilon}, \quad (9)$$

where μ_g and σ_g are the mean and standard deviation of trajectory returns within the rollout group.

We then optimize a GRPO-style clipped surrogate objective with KL regularization:

$$\begin{aligned}\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\Big(&\rho_{i,t}(\theta)A_{i,t},\\ &\text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)A_{i,t}\Big)\\ &+\beta\,\text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big).\end{aligned} \quad (10)$$

where

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid x_i, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid x_i, o_{i,<t})}. \quad (11)$$

Notably, the process reward is computed with the same frozen reference model π_{ref} used in KL regularization. Thus, PAMT does not require any additional scoring model. The full RL training procedure is summarized in Algorithm 1 and a detailed derivation from standard GRPO to our process-aware formulation is provided in Appendix E.

5 Experiments

5.1 Experimental Settings

Datasets. We use two training sets for the two stages of PAMT: a curated 7K domain-aware Long-CoT dataset for cold-start SFT, and a separate 20K multi-domain parallel dataset for RL. Evaluation covers in-domain, OOD, and multilingual test sets over seen and unseen language pairs. Full details are provided in Appendix B.1 and Appendix A.1.

Metrics. We evaluate translation quality with BLEU, COMET (Rei et al., 2020), and COMETKiwi (Rei et al., 2022b). Metric details and implementation settings are respectively provided in Appendix B.6 and Appendix A.2.

Baselines. To isolate the effects of reasoning and process alignment, we compare PAMT with three groups of systems: ***LLMs***, which treat

Model	Laws			News			Science			Subtitles		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI
Large Language Models												
DeepSeek-V3	59.58	89.15	84.42	35.86	86.94	85.44	33.54	88.28	85.46	23.66	82.44	82.69
Gemini-2.0-Flash	56.43	88.90	84.40	36.39	86.37	85.14	33.47	87.69	85.24	24.64	81.91	82.33
GPT-4o	48.50	88.28	84.52	34.58	86.22	85.02	32.09	87.34	84.84	24.91	81.91	81.93
Large Reasoning Models												
DeepSeek-R1	59.34	89.15	84.68	33.35	86.67	85.40	31.82	88.08	85.47	20.50	82.10	82.84
Gemini-2.0-Flash-Thinking	55.75	88.61	84.27	33.90	86.02	84.79	31.98	87.47	85.06	23.83	80.90	81.22
GPT-5	54.07	89.34	85.43	35.05	86.65	85.52	32.21	88.06	85.47	23.23	82.32	82.68
MT-Specialized Models												
SFT-Parallel-7B	56.64	**88.83**	84.29	28.89	85.43	83.91	29.28	86.19	83.68	26.93	81.41	80.26
ALMA-7B-R	36.34	85.45	81.87	24.53	83.95	81.64	22.16	84.33	81.83	18.56	80.18	79.40
ALMA-13B-R	40.33	86.89	83.10	26.87	84.65	82.43	24.37	85.72	82.59	19.54	81.11	80.00
TowerInstruct-7B-v0.2	50.54	88.25	82.96	30.72	84.61	82.45	27.76	85.60	82.99	22.45	80.75	80.01
MT-RewardTree	38.73	85.45	83.16	30.53	85.38	83.92	30.23	86.90	84.59	20.83	80.60	82.61
CoT-FT-7B	**57.05**	88.77	84.35	30.17	85.23	83.31	28.72	86.02	83.48	**28.17**	80.92	79.20
MT-R1-Zero-7B	35.50	86.79	84.51	31.69	86.01	84.54	29.45	86.87	84.61	22.48	**81.73**	81.70
SSR-X-Zero-7B	38.88	86.44	83.44	28.14	85.87	83.98	28.54	87.11	84.65	23.18	80.50	81.99
mExTrans-7B	38.78	87.11	84.44	25.05	86.07	**85.34**	25.74	**87.34**	**85.31**	14.31	81.40	**82.85**
TAT-R1	44.64	87.21	83.90	31.76	85.90	84.49	29.95	87.10	84.72	24.65	80.36	82.07
Our Models												
PAMT-Qwen2.5-7B-Instruct	52.11	88.58	84.88	**33.50**	**86.11**	84.82	**32.02**	87.25	84.98	27.41	80.87	82.62
PAMT-Gemma2-9B-IT	51.31	88.50	**84.90**	32.96	86.00	84.81	31.11	86.75	84.59	27.37	80.97	82.57

Model	Literary			IT			Koran			Medical			Average		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI
Large Language Models															
DeepSeek-V3	16.14	77.28	77.07	38.16	83.90	78.86	17.94	74.91	80.33	41.43	84.15	81.90	33.29	83.38	82.02
Gemini-2.0-Flash	18.37	77.20	76.67	37.93	83.20	78.49	19.70	74.97	79.71	44.39	84.49	81.76	33.92	83.09	81.72
GPT-4o	17.75	77.48	77.18	37.23	83.52	78.37	17.65	75.04	80.64	41.89	84.23	81.89	31.83	83.00	81.80
Large Reasoning Models															
DeepSeek-R1	11.25	75.44	75.16	36.66	83.49	78.68	17.05	74.76	80.63	40.69	83.94	82.39	31.33	82.95	81.91
Gemini-2.0-Flash-Thinking	18.01	77.06	76.63	37.26	82.94	78.40	19.57	75.03	79.79	43.36	84.14	81.70	32.96	82.77	81.48
GPT-5	15.01	76.72	77.08	36.92	83.58	79.22	18.83	75.49	81.80	43.00	84.29	83.13	32.29	83.31	82.54
MT-Specialized Models															
SFT-Parallel-7B	15.77	76.73	75.37	**40.64**	**84.08**	79.35	**21.09**	74.66	78.39	43.76	84.40	82.47	32.88	82.72	80.96
ALMA-7B-R	13.46	75.12	74.97	33.75	81.12	77.72	14.09	71.92	79.12	37.97	83.04	81.36	25.11	80.64	79.74
ALMA-13B-R	14.20	75.89	75.81	34.22	81.63	77.78	14.75	72.76	79.63	40.34	83.38	81.49	26.83	81.50	80.35
TowerInstruct-7B-v0.2	15.54	75.49	74.93	38.20	83.47	78.67	10.99	69.02	70.13	**46.81**	84.58	80.80	30.38	81.47	79.12
MT-RewardTree	14.22	76.27	77.29	31.53	80.41	77.64	14.23	73.39	80.92	32.30	81.59	81.21	26.57	81.25	81.42
CoT-FT-7B	15.31	76.45	74.96	40.41	83.79	79.27	20.52	74.04	78.16	44.21	84.34	82.59	33.07	82.44	80.67
MT-R1-Zero-7B	13.75	76.93	76.72	34.51	82.71	79.59	13.05	72.84	80.66	27.28	83.31	82.69	25.96	82.15	81.88
SSR-X-Zero-7B	13.66	76.82	76.14	27.01	80.08	76.57	13.97	72.86	79.75	28.86	82.12	80.98	25.28	81.47	80.94
mExTrans-7B	10.43	76.16	76.62	25.54	78.76	77.19	11.80	73.88	80.56	25.40	81.66	81.93	22.13	81.55	81.78
TAT-R1	16.63	77.50	76.80	34.43	81.57	77.67	14.79	73.29	80.61	37.39	83.16	81.71	29.28	82.01	81.50
Our Models															
PAMT-Qwen2.5-7B-Instruct	16.97	77.96	77.26	39.11	83.75	**80.03**	18.31	74.94	**81.10**	42.12	84.13	82.99	32.69	82.95	**82.33**
PAMT-Gemma2-9B-IT	**18.06**	**78.24**	**77.61**	40.00	83.85	79.96	19.91	**75.46**	80.80	44.65	**84.61**	**83.06**	**33.17**	**83.05**	82.29

Table 4: In-domain performance on 8 domains, measured by BLEU, COMET, and COMETKiwi (KIWI). Within MT-Specialized Models and Our Models, the best score is shown in bold and the second-best is underlined.

MT mainly as next-token generation, including DeepSeek-V3 (DeepSeek-AI et al., 2025), Gemini-2.0-Flash (DeepMind, 2024), GPT-4o (OpenAI et al., 2024), Gemma2-9B-IT (Gemma Team et al., 2024), and Qwen2.5-7B-Instruct (Yang et al., 2025a). ***LRMs***, which expose explicit reasoning but are not designed for MT process alignment, including DeepSeek-R1 (Guo et al., 2025), Gemini-2.0-Flash-Thinking (DeepMind, 2025), and GPT-5 (OpenAI, 2025). ***MT-specialized models***, including non-reasoning MT systems such as TowerInstruct (Alves et al., 2024), ALMA-R (Xu et al., 2024a,b), SFT-Parallel, Tower-Plus-9B (Rei et al., 2025), and MT-RewardTree (Feng et al., 2025b), as well as prior reasoning-augmented MT methods including CoT-FT (Hu et al., 2024), MT-R1-Zero (Feng et al., 2025a), mExTrans (Wang et al., 2025d), SSR-X-Zero (Yang et al., 2025b), and TAT-R1 (Li et al., 2025).

5.2 Generalization Across Domains

A key question for process-aligned MT is whether it learns reusable translation behavior or merely overfits to domain-specific heuristics. As shown in Tables 4 and 5, PAMT consistently outperforms prior MT-specialized baselines on both in-domain and OOD evaluations, achieving stronger average BLEU, COMET, and COMETKiwi scores. In contrast, CoT-FT relies on offline CoT imitation and therefore generalizes less reliably under

Model	Conversation			E-commerce			Social			Culture			CommonSense			Average		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI
Large Language Models																		
DeepSeek-V3	36.76	87.03	81.48	32.29	85.53	81.19	32.26	84.59	81.44	40.27	85.46	83.21	32.55	85.36	79.98	34.83	85.59	81.46
Gemini-2.0-Flash	38.05	86.90	81.37	32.41	85.49	80.87	32.98	84.28	81.04	39.11	85.02	82.94	31.60	84.71	79.38	34.83	85.28	81.12
GPT-4o	38.02	86.79	81.43	32.88	85.57	81.04	32.75	84.47	81.08	38.65	85.24	83.14	32.54	85.26	79.87	34.97	85.47	81.31
Large Reasoning Models																		
DeepSeek-R1	33.34	86.47	81.37	27.41	85.03	80.91	27.22	83.94	81.17	35.88	85.45	83.14	28.50	84.52	79.99	30.47	85.08	81.32
Gemini-2.0-Flash-Thinking	37.21	86.64	81.24	31.50	85.24	80.74	31.79	83.89	80.83	36.94	84.63	83.68	32.15	84.50	79.68	33.92	84.98	81.23
GPT-5	35.25	87.14	82.82	29.09	85.37	82.02	29.00	84.10	82.31	35.79	85.04	84.15	27.08	84.43	80.75	31.24	85.22	82.41
MT-Specialized Models																		
SFT-Parallel-7B	32.20	84.11	80.62	26.94	82.92	80.12	25.88	81.61	80.12	31.81	82.87	81.35	22.19	81.91	78.98	27.81	82.68	80.24
ALMA-7B-R	29.49	84.67	79.96	25.13	82.98	79.50	25.76	82.51	79.77	33.25	83.81	82.83	23.25	82.80	80.00	27.38	83.35	80.41
MT-RewardTree	31.34	85.59	81.82	27.32	84.04	81.28	26.31	83.14	**81.80**	31.30	82.79	81.69	22.59	82.86	79.94	27.77	83.68	81.31
CoT-FT-7B	31.58	84.09	80.63	26.71	82.95	80.11	24.98	81.37	79.83	30.41	82.46	81.10	22.38	81.97	78.88	27.21	82.57	80.11
MT-R1-Zero-7B	32.38	85.88	81.50	26.98	84.25	81.25	26.39	83.28	81.40	31.86	83.99	82.85	24.50	82.94	79.53	28.42	84.07	81.31
SSR-X-Zero-7B	30.65	85.32	81.13	25.64	83.94	80.79	25.47	83.19	81.26	28.70	83.25	81.11	23.93	82.82	79.80	26.88	83.70	80.82
mExTrans-7B	24.53	84.72	81.15	20.46	83.52	81.24	19.60	82.29	81.70	27.12	**84.30**	**83.92**	18.61	81.92	79.23	22.06	83.35	81.45
TAT-R1	34.18	85.85	81.20	29.77	84.50	81.05	29.37	83.51	81.27	35.36	84.17	83.27	28.18	84.02	79.62	31.37	84.41	81.28
Our Models																		
PAMT-Qwen2.5-7B-Instruct	34.77	86.11	81.88	**31.20**	84.85	81.60	**31.10**	83.70	81.66	**36.15**	84.23	83.62	30.36	**84.30**	80.07	**32.72**	**84.64**	81.77
PAMT-Gemma2-9B-IT	**35.22**	**86.13**	**82.05**	30.30	**84.93**	**81.66**	30.98	**83.73**	81.60	34.53	84.02	83.59	**31.17**	83.81	**80.22**	32.44	84.52	**81.83**

Table 5: Out-of-domain performance on 5 domains, measured by BLEU, COMET, and KIWI.

Model	En⇒Zh			Zh⇒En			De⇒En			En⇒X			X⇒En			Average		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI	BLEU	COMET	KIWI
MT-Specialized Models																		
ALMA-7B-R	26.69	84.13	81.43	13.46	75.12	74.97	28.87	79.66	80.13	2.78	58.13	73.13	12.01	63.94	58.17	16.76	72.20	73.57
Tower-Plus-9B	37.91	**86.76**	84.88	16.56	77.88	77.78	**36.56**	82.21	**82.36**	6.07	63.86	69.46	27.99	81.38	79.68	25.02	78.42	78.83
SFT-Parallel-7B	38.58	86.34	83.40	15.77	76.73	75.37	35.72	81.63	81.11	1.19	52.16	43.97	18.47	75.58	73.05	21.95	74.49	71.38
MT-R1-Zero-7B	32.00	86.23	84.35	13.75	76.93	76.72	25.56	80.39	81.73	5.91	58.76	58.27	21.91	77.48	75.61	19.82	75.96	75.34
SSR-X-Zero-7B	31.00	85.41	82.55	19.46	80.38	78.18	26.53	81.79	81.09	6.02	58.94	56.34	20.73	78.05	75.78	20.75	76.92	74.79
mExTrans-7B	28.63	86.44	**85.02**	10.43	76.16	76.62	22.06	79.27	80.88	5.09	61.33	61.74	15.95	76.80	75.21	16.43	76.00	75.89
Our Models																		
PAMT-Qwen2.5-7B-Instruct	**38.66**	86.05	83.75	24.02	**81.68**	79.01	32.50	83.16	82.26	5.90	58.15	54.02	22.08	76.86	75.44	24.63	77.18	74.89
PAMT-Gemma2-9B-IT	37.11	85.87	83.68	**25.01**	81.54	**79.20**	33.23	**83.51**	82.31	**8.63**	**66.69**	**82.87**	**32.91**	**84.92**	**83.52**	**27.38**	**80.51**	**82.31**

Table 6: Multilingual performance on 5 language settings, measured by BLEU, COMET, and KIWI.

domain shift, while TAT-R1 improves robustness with terminology-aware rewards but still operates mainly at the output level. As a result, neither method can identify which intermediate translation step leads to correct domain-sensitive choices. By attributing changes in final translation likelihood to individual translation steps, PAMT enables step-level credit assignment, reinforcing the decisions that transfer across domains and suppressing those that cause process drift.

5.3 Generalization Across Languages

Table 6 reports the multilingual results, which test whether the model learns language-pair-specific patterns or a translation process that transfers across directions. PAMT achieves the strongest average multilingual performance among MT-specialized baselines and our models, with gains that are especially clear in the transfer-heavy En-X and X-En settings. This pattern suggests that PAMT learns more than language-specific templates. We attribute this advantage to step-level process alignment: sequence-level rewards collapse source interpretation, structural transfer, and target-language realization into a single terminal signal, whereas PAMT credits each explicit translation step by its effect on the reference translation likelihood. This allows the model to reinforce transferable intermediate decisions while suppressing brittle language-specific shortcuts, leading to stronger generalization across language directions.

Method	In-Domain			Out-of-Domain		
	BLEU	COMET	KIWI	BLEU	COMET	KIWI
PAMT	29.83	82.10	81.31	32.72	84.64	81.77
PAMT w/o credit assign	29.33	81.99	81.04	31.62	84.12	81.15
PAMT w/o process reward	28.51	81.79	81.14	32.05	84.33	81.41
PAMT w/o quality reward	24.94	81.01	80.55	26.50	83.42	81.14
PAMT w/o RL	24.45	80.77	80.59	25.10	83.50	81.17

Table 7: Ablation study on in-domain and out-of-domain test sets.

5.4 Ablation Study

Table 7 shows that all components of PAMT contribute to the final gains. Removing credit assignment consistently degrades performance in both in-domain and out-of-domain settings, indicating that step-level process gains must be propagated

Model	Acc. (%) ↓	Fluency (%) ↓	Non-trans (%) ↓
CoT-FT-7B	18.54	19.19	13.19
SFT-Parallel-7B	18.29	18.41	8.79
TAT-R1	18.11	19.81	46.15
PAMT w/o Process	14.93	15.19	12.09
PAMT-Qwen2.5-7B-Instruct	**14.80**	14.41	**6.59**
PAMT-Gemma2-9B-IT	15.32	**12.99**	13.19
Model	**Style (%) ↓**	**Term. (%) ↓**	**Term. Acc. (%) ↑**
CoT-FT-7B	17.58	16.89	40.44
SFT-Parallel-7B	16.76	16.82	40.03
TAT-R1	19.33	19.19	40.95
PAMT w/o Process	16.25	16.64	41.72
PAMT-Qwen2.5-7B-Instruct	15.93	16.20	41.88
PAMT-Gemma2-9B-IT	**14.15**	**14.26**	**42.97**

Table 8: MQM (↓) and terminology accuracy (↑) results. The top section reports MQM error rates for accuracy (Acc.), fluency, and non-translation (Non-trans), while the bottom section reports style, terminology (Term.), and terminology accuracy (Term. Acc.).

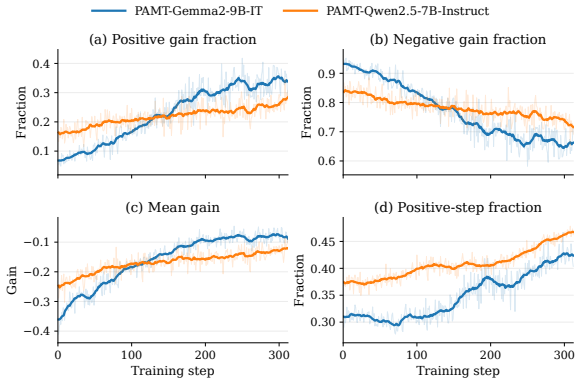

Figure 2: Training dynamics of process rewards

to the corresponding reasoning tokens to become effective optimization signals. Removing the process reward also leads to clear drops, suggesting that sequence-level outcome supervision alone is insufficient for improving intermediate translation reasoning. The largest degradation comes from removing the quality reward, confirming that explicit supervision on the final translation remains indispensable. Finally, removing RL yields the weakest overall results, showing that supervised initialization alone cannot fully translate explicit reasoning traces into translation gains. Overall, these results validate that PAMT relies on the synergy of quality reward, process reward, and token-level credit assignment.

5.5 PAMT on Terminology and Style Drift

Table 8 directly evaluates whether process supervision mitigates terminology and style drift. PAMT achieves the lowest error rates overall. Compared with CoT-FT, PAMT-Gemma2-9B-IT reduces style and terminology errors by 3.43 and 2.63 points, showing that exposing the process alone is insufficient. The contrast with TAT-R1 is more striking: despite explicit terminology constraints, it exhibits substantially higher style and terminology errors (19.33/19.19 vs. 14.15/14.26) and an extremely high non-translation rate (46.15), indicating that output-level constraints cannot stabilize an unaligned process. The ablation further confirms the role of process reward: removing it increases style and terminology errors (from 14.15/14.26 to 16.25/16.64). Notably, PAMT also achieves higher terminology accuracy than TAT-R1 (42.97 vs. 40.95). These results show that terminology and style drift arise from misaligned intermediate translation steps, and that PAMT mitigates this bottleneck through step-level credit assignment. We further provide qualitative case studies in Appendix D.

5.6 Training Dynamics of Process Rewards

Figure 2 shows that PAMT induces highly consistent process-level learning dynamics across two backbones. As training proceeds, the fraction of positive-gain reasoning steps steadily increases, while the fraction of negative-gain steps decreases. Meanwhile, the mean process gain also improves, indicating that intermediate reasoning steps become progressively less harmful and more supportive of the reference translation. We further observe that the positive-step fraction within each trajectory rises over time, suggesting that the improvement is distributed across the reasoning process rather than concentrated in only a few isolated steps. Overall, these trends provide direct evidence that PAMT effectively optimizes intermediate translation reasoning, rather than only improving the final output.

6 Conclusion

In this work, we show that explicit translation reasoning in MDMT is both a strength and a liability: it improves long-form and high-difficulty translation, but often drifts in terminology-intensive and stylistically constrained settings when intermediate decisions are not properly supervised. To address this, we propose PAMT, a process-aligned training framework that breaks the credit-assignment bottleneck by tying each explicit translation step to its contribution to the final translation. Across two backbones, PAMT consistently improves in-domain, OOD, and multilingual translation, outperforming MT-specialized baselines while remaining competitive with strong LLM/LRM systems.

Limitation

Our study has several limitations. PAMT is designed for explicit translation processes, i.e., intermediate translation steps verbalized in the `<think>` field, and therefore does not directly model latent internal reasoning that is not exposed in the output trace. In addition, our process reward is reference-based during training, which provides a stable signal for fine-grained credit assignment but assumes access to parallel data. The current implementation also uses simple delimiter-based step segmentation and uniform reward distribution within each step, which is effective in practice but still a coarse approximation of step-level contribution. Finally, process-aware optimization introduces additional training-time cost because multiple step prefixes must be evaluated for each sampled trace, although this does not affect inference. Extending PAMT to weaker-supervision settings, more adaptive process segmentation, and more efficient training remains important future work.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *Preprint*, arXiv:2402.17733.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Eleftheria Briakou, Jiaming Luo, Colin Cherry, and Markus Freitag. 2024. Translating step-by-step: Decomposing the translation process for improved translation quality of long-form texts. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1301–1317, Miami, Florida, USA. Association for Computational Linguistics.

Andong Chen, Lianzhang Lou, Kehai Chen, Xuefeng Bai, Yang Xiang, Muyun Yang, Tiejun Zhao, and Min Zhang. 2024a. DUAL-REFLECT: Enhancing large language models for reflective translation through dual learning feedback mechanisms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 693–704, Bangkok, Thailand. Association for Computational Linguistics.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2024b. Iterative translation refinement with large language models. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 181–190, Sheffield, UK. European Association for Machine Translation (EAMT).

Google DeepMind. 2024. Introducing gemini 2.0: our new ai model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#ceo-message. Accessed: 2025-04-21.

Google DeepMind. 2025. Gemini 2.0 flash thinking. https://deepmind.google/technologies/gemini/flash-thinking. Accessed: 2025-04-21.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Zhaopeng Feng, Shaosheng Cao, Jiahan Ren, Jiayuan Su, Ruizhe Chen, Yan Zhang, Jian Wu, and Zuozhu Liu. 2025a. MT-r1-zero: Advancing LLM-based machine translation via r1-zero-like reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 18685–18702, Suzhou, China. Association for Computational Linguistics.

Zhaopeng Feng, Jiahan Ren, Jiayuan Su, Jiamei Zheng, Hongwei Wang, and Zuozhu Liu. 2025b. Mt-rewardtree: A comprehensive framework for advancing llm-based machine translation via reward modeling. *Preprint*, arXiv:2503.12123.

Zhaopeng Feng, Yan Zhang, Hao Li, Bei Wu, Jiayu Liao, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2025c. TEaR: Improving LLM-based machine translation with systematic self-refinement. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3922–3938, Albuquerque, New Mexico. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Gemma Team, Morgane Riviere, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 180 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jiaxin Guo, Hao Yang, Zongyao Li, Daimeng Wei, Hengchao Shang, and Xiaoyu Chen. 2024. A novel paradigm boosting translation capabilities of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 639–649, Mexico City, Mexico. Association for Computational Linguistics.

Jie He, Tao Wang, Deyi Xiong, and Qun Liu. 2020. The box is in the pen: Evaluating commonsense reasoning in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3662–3672, Online. Association for Computational Linguistics.

Minggui He, Yilun Liu, Shimin Tao, Yuanchang Luo, Hongyong Zeng, Chang Su, Li Zhang, Hongxia Ma, Daimeng Wei, Weibin Meng, Hao Yang, Boxing Chen, and Osamu Yoshie. 2025. R1-t1: Fully incentivizing translation capability in llms via reasoning learning. *Preprint*, arXiv:2502.19735.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024. Exploring human-like translation strategy with large language models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

Tianxiang Hu, Pei Zhang, Baosong Yang, Jun Xie, Derek F. Wong, and Rui Wang. 2024. Large language model for multi-domain translation: Benchmarking and domain CoT fine-tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5726–5746, Miami, Florida, USA. Association for Computational Linguistics.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1834, Online. Association for Computational Linguistics.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4253–4273, Mexico City, Mexico. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Wen Lai, Alexandra Chronopoulou, and Alexander Fraser. 2022. m^4 adapter: Multilingual multi-domain adaptation for machine translation with a meta-adapter. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4282–4296, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zheng Li, Mao Zheng, Mingyang Song, and Wenjie Yang. 2025. Tat-r1: Terminology-aware translation with reinforcement learning and word alignment. *Preprint*, arXiv:2505.21172.

Zhibo Man, Yuanmeng Chen, Yujie Zhang, Yufeng Chen, and Jinan Xu. 2025. Dmdteval: An evaluation and analysis of llms on disambiguation in multi-domain translation. *arXiv preprint arXiv:2504.20371*.

Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the WMT 2018 biomedical translation shared task: Evaluation on Medline test sets. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 324–339, Belgium, Brussels. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. Introducing GPT-5. https://openai.com/zh-Hans-CN/index/introducing-gpt-5/.

Jianhui Pang, Fanghua Ye, Derek Fai Wong, Dian Yu, Shuming Shi, Zhaopeng Tu, and Longyue Wang. 2025. Salute the classic: Revisiting challenges of machine translation in the age of large language models. *Transactions of the Association for Computational Linguistics*, 13:73–95.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022a. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, João Alves, Pedro Teixeirinha, Amin Farajian, and André F. T. Martins. 2025. Tower+: Bridging generality and translation specialization in multilingual llms. *Preprint*, arXiv:2506.17080.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Danielle Saunders. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424.

Kirill Semenov, Vilém Zouhar, Tom Kocmi, Dongdong Zhang, Wangchunshu Zhou, and Yuchen Eleanor Jiang. 2023. Findings of the WMT 2023 shared task on machine translation with terminologies. In *Proceedings of the Eighth Conference on Machine Translation*, pages 663–671, Singapore. Association for Computational Linguistics.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.

Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. 2014. UM-corpus: A large English-Chinese parallel corpus for statistical machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1837–1842, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jiaan Wang, Fandong Meng, Yunlong Liang, and Jie Zhou. 2025a. Drt: Deep reasoning translation via long chain-of-thought. *Preprint*, arXiv:2412.17498.

Jiaan Wang, Fandong Meng, Yunlong Liang, and Jie Zhou. 2025b. DRT: Deep reasoning translation via long chain-of-thought. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6770–6782, Vienna, Austria. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, and Jie Zhou. 2025c. Deep reasoning translation via reinforcement learning. *Preprint*, arXiv:2504.10187.

Jiaan Wang, Fandong Meng, and Jie Zhou. 2025d. Extrans: Multilingual deep reasoning translation via exemplar-enhanced reinforcement learning. *Preprint*, arXiv:2505.12996.

Jiaan Wang, Fandong Meng, and Jie Zhou. 2026. DeepTrans: Deep reasoning translation via reinforcement learning. *Transactions of the Association for Computational Linguistics*, 14:47–63.

Longyue Wang, Siyou Liu, Chenyang Lyu, Wenxiang Jiao, Xing Wang, Jiahao Xu, Zhaopeng Tu, Yan Gu, Weiyu Chen, Minghao Wu, Liting Zhou, Philipp Koehn, Andy Way, and Yulin Yuan. 2024a. Findings of the WMT 2024 shared task on discourse-level literary translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 699–700, Miami, Florida, USA. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Yan Gu, Siyou Liu, Dian Yu, Qingsong Ma, Chenyang Lyu, Liting Zhou, Chao-Hong Liu, Yufeng Ma, Weiyu Chen, Yvette Graham, Bonnie Webber, Philipp Koehn, Andy Way, Yulin Yuan, and Shuming Shi. 2023. Findings of the WMT 2023 shared task on discourse-level literary translation: A fresh orb in the cosmos of LLMs. In *Proceedings of the Eighth Conference on Machine Translation*, pages 55–67, Singapore. Association for Computational Linguistics.

Yutong Wang, Jiali Zeng, Xuebo Liu, Fandong Meng, Jie Zhou, and Min Zhang. 2024b. TasTe: Teaching large language models to translate through self-reflection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6144–6158,

Bangkok, Thailand. Association for Computational Linguistics.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024a. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

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. In *Forty-first International Conference on Machine Learning*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025a. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Wenjie Yang, Mao Zheng, Mingyang Song, Zheng Li, and Sitong Wang. 2025b. Ssr-zero: Simple self-rewarding reinforcement learning for machine translation. *Preprint*, arXiv:2505.16637.

Binwei Yao, Ming Jiang, Tara Bobinac, Diyi Yang, and Junjie Hu. 2024. Benchmarking machine translation with cultural awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13078–13096, Miami, Florida, USA. Association for Computational Linguistics.

Jiawei Zheng, Hanghai Hong, Feiyan Liu, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. 2024a. Fine-tuning large language models for domain-specific machine translation. *arXiv preprint arXiv:2402.15061*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024b. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Training Setup

A.1 Training Data

Cold-start SFT data. The cold-start supervised fine-tuning (SFT) stage uses a curated Long-CoT translation dataset spanning ten domains and three translation directions: De→En, En→Zh, and Zh→En. The dataset contains approximately 7K examples in total. Each example follows the structured format `<think>... </think><answer>... </answer>`, where the `<think>` segment describes an explicit translation process and the `<answer>` segment provides the final translation. These examples are distilled from strong teacher models to initialize the model's ability to externalize translation reasoning.

RL training data. For reinforcement learning (RL), we construct a multilingual multi-domain machine translation training set covering three directions: De→En, En→Zh, and Zh→En. The De→En portion is drawn from the multi-domain dataset of Aharoni and Goldberg (2020), covering IT, Law, Medical, Koran, and Subtitles. The En→Zh portion is drawn from UM-Corpus (Tian et al., 2014), covering News, Laws, Subtitles, and Science. The Zh→En Literary domain is taken from the GuoFeng-Webnovel dataset used in the WMT23 and WMT24 literary translation tasks (Wang et al., 2023, 2024a). From each domain, we randomly sample 2K sentence pairs and retain only examples with a minimum source length of 20 words, or 20 Chinese characters for Chinese inputs, to ensure sufficient room for explicit translation reasoning. The final RL training set contains 20K examples.

A.2 Implementation Details

Cold-start SFT. Our SFT implementation is based on LLaMA-Factory[1] (Zheng et al., 2024b). We adopt Qwen-2.5-7B-Instruct and Gemma2-9B-IT as the base models and fine-tune them on the 7K difficulty-adaptive Long-CoT examples with full-parameter optimization. Training is conducted for 2 epochs on 8 NVIDIA A100 80GB GPUs with a global batch size of 32. We use the AdamW optimizer with a learning rate of $1e{-5}$, a cosine learning rate scheduler, and a warm-up ratio of 0.1. The maximum input sequence length is set to 4096 tokens. We apply DeepSpeed ZeRO Stage 3 for memory-efficient training.

RL training. Our RL implementation is built on `verl`[2] (Sheng et al., 2025). We train the model for 2 epochs on 8 NVIDIA A100 80GB GPUs with a global batch size of 128. The rollout number is set to 8, and the rollout temperature is set to 1.0. For PPO updates, the optimization mini-batch size is 16. The learning rate is set to $1e{-6}$, the KL loss coefficient is $\beta = 1e{-3}$, and the process reward weight is $\lambda = 0.1$. The maximum response length is set to 2048 tokens. The entire RL training stage takes about 9 hours.

[1] https://github.com/hiyouga/LLaMA-Factory
[2] https://github.com/volcengine/verl

Inference. During inference, we use the vLLM[3] backend (Kwon et al., 2023) for efficient decoding, with temperature set to 0.0 and a repetition penalty of 1.05.

B Evaluation Setup

We use the same evaluation setup for both the preliminary analysis in Section 2 and the main experiments in Section 5. This section describes the multi-domain benchmark, the in-domain and out-of-domain test sets, the multilingual evaluation protocol, and the post-training data scale of MT-specialized baselines.

B.1 Multi-domain Evaluation Benchmark

Scope and design. We build a unified multi-domain benchmark from publicly available corpora with explicit splits. The benchmark serves two purposes. First, it supports the diagnostic analyses in Section 2 by providing broad coverage across four translation directions: De⇒En, En⇒De, En⇒Zh, and Zh⇒En. Second, it provides the in-domain and out-of-domain evaluation sets used in the main experiments. The benchmark spans both general-purpose and domain-specific text, including biomedical, legal, IT, science, subtitles, conversation, social media, cultural, commonsense, and literary content. It is designed to cover high-resource and low-resource conditions, terminology-intensive domains, context-sensitive inputs, stylistically demanding text, and noisier informal genres, while minimizing potential data leakage and preserving comparability across systems.

Data sources. For German⇔English, the benchmark covers 11 domains. Five domains—Medical, Law, IT, Koran, and Subtitles—are taken from the publicly available Multi-Domain dataset (Aharoni and Goldberg, 2020). The Mixed domain is drawn from the WMT22 General Machine Translation Task (Kocmi et al., 2022), whose subdomains include News, Social, E-commerce, and Conversation. Biomedical data is collected from the WMT18 and WMT19 Biomedical Machine Translation Tasks (Neves et al., 2018; Bawden et al., 2019).

For Chinese⇔English, the benchmark covers 12 domains. The Commonsense and Culture domains are taken from CommonMT (He et al., 2020) and CAMT (Yao et al., 2024), respectively, while the Literary domain is taken from the WMT23 Literary Machine Translation Task (Wang et al., 2023). The general-purpose Mixed domain, together with its News, Social, E-commerce, and Conversation subdomains, is again sourced from the WMT22 General Machine Translation Task (Kocmi et al., 2022). Biomedical data is collected from the WMT18 and WMT19 Biomedical Machine Translation Tasks (Neves et al., 2018; Bawden et al., 2019). In addition, the Laws, News, Science, and Subtitles domains are drawn from UM-Corpus (Tian et al., 2014).

[3] https://github.com/vllm-project/vllm

Test set	Num	Test set	Num
Conversation	349	Social	491
E-commerce	518	News (WMT22)	499
Bio (WMT18)	212	Bio (WMT19)	243
Commonsense	1200	Literary	3038
Terminology	1638	Mixed (WMT22)	1875

Table 9: Test sets and the number of samples for Zh⇒En translation tasks. Bio denotes the Biomedical domain.

Test set	Num	Test set	Num
Culture	778	Laws	456
Subtitles	597	Science	503
Conversation	484	Social	511
E-commerce	530	News (WMT22)	511
Bio (WMT18)	244	Bio (WMT19)	224
News (UM)	1500	Mixed	2037

Table 10: Test sets and the number of samples for En⇒Zh translation tasks.

Detailed statistics. Tables 9–12 report the number of examples for each domain and direction. No additional filtering or augmentation is applied to these evaluation sets.

B.2 In-Domain Test Sets

For in-domain evaluation, we use the official test sets associated with the corpora used for RL training. Specifically, these sets are drawn from the German-English multi-domain dataset (Aharoni and Goldberg, 2020), UM-Corpus (Tian et al., 2014), and the GuoFeng-Webnovel literary dataset (Wang et al., 2023, 2024a). For the Literary domain, we merge the `valid_1`, `valid_2`, `test_1`, and `test_2` splits into a single evaluation set. Table 13 summarizes the in-domain test sets used for the three training directions.

Test set	Num	Test set	Num
Bio (WMT18)	217	Bio (WMT19)	373
Social	515	E-commerce	501
News (WMT22)	506	Conversation	462
IT	2000	Law	2000
Koran	2000	Subtitles	2000
Medical	2000	Terminology	2955
Mixed	1984		

Table 11: Test sets and the number of samples for De⇒En translation tasks.

Test set	Num	Test set	Num
Bio (WMT18)	220	Bio (WMT19)	435
News (WMT22)	511	E-commerce	530
Social	512	Conversation	484
Mixed	2037		

Table 12: Test sets and the number of samples for En⇒De translation tasks.

Domain	Num	Domain	Num
En→Zh Laws	456	De→En IT	2000
En→Zh Subtitles	597	De→En Koran	2000
En→Zh Science	503	De→En Medical	2000
En→Zh News	1500	De→En Law	2000
Zh→En Literary	3038	De→En Subtitles	2000

Table 13: In-domain test sets and the number of samples for En↔Zh and De→En translation tasks.

Domain	Num	Domain	Num
En→Zh Conversation	484	En→Zh Social	511
En→Zh Ecommerce	530	En→Zh Culture	778
Zh→En CommonSense	1200	Zh→En Conversation	349
Zh→En Social	491	Zh→En Ecommerce	518
De→En Conversation	462	De→En Social	515
De→En Ecommerce	501		

Table 14: Out-of-domain test sets and sample counts for En↔Zh and De→En translation tasks.

B.3 Out-of-Domain Test Sets

To evaluate cross-domain generalization, we use public test sets from domains that are not included in the RL training data. Specifically, the Conversation, E-commerce, and Social domains are taken from the WMT22 shared tasks (Kocmi et al., 2022), the Culture domain is drawn from CAMT (Yao et al., 2024), and the Commonsense domain comes from CommonMT (He et al., 2020). These test sets complement the in-domain evaluation by introducing more informal, culturally grounded, and knowledge-sensitive inputs. Table 14 reports the corresponding statistics.

B.4 Multilingual Evaluation

For unseen-language evaluation, we use the FLORES+ benchmark (NLLB Team et al., 2024) and construct an *unseen* language set to minimize leakage from languages already covered by baseline post-training data (Table 6). We first remove all languages appearing in baseline training coverage, including Chinese (zh), English (en), German (de), French (fr), Spanish (es), Portuguese (pt), Italian (it), Russian (ru), Korean (ko), Dutch (nl), Czech (cs), Icelandic (is), Ukrainian (uk), Hindi (hi), Japanese (ja), Polish (pl), Swedish (sv), Hungarian (hu), Romanian (ro), Danish (da), Norwegian (no), and Finnish (fi). We then further restrict the remaining FLORES+ languages to those supported by both COMET and COMETKIWI, so that evaluation is consistent across all unseen directions. After these two filtering steps, the final unseen-language set contains 59 languages, listed in Table 15.

The *seen* languages in our setting are German (de), English (en), and Chinese (zh), which appear in our training data through the German-English multi-domain dataset (Aharoni and Goldberg, 2020), UM-Corpus (Tian et al., 2014), and the GuoFeng-Webnovel literary dataset.

B.5 Training Data Scale of MT Baselines

To facilitate fair comparison, we report the post-training data scale used by each MT-specialized baseline. PAMT and most reasoning-augmented baselines, including MT-R1-Zero-7B, CoT-FT-7B, SFT-Parallel, and mExTrans-7B, are trained on roughly 27K examples. In contrast, several other baselines use substantially larger corpora: TowerInstruct uses 637K examples, Tower-Plus-9B uses 286K, and ALMA-R uses 21K. SSR-X-Zero-7B is trained on a smaller subset of 13K instances.

B.6 Evaluation Metrics

Automatic Metrics. We use three automatic evaluation metrics:

- **BLEU**[4] (Post, 2018), which evaluates surface-level n-gram overlap between system output and reference translations.
- **COMET**[5] (Rei et al., 2022a), a reference-based semantic metric trained on human quality judgments.

[4] https://github.com/mjpost/sacrebleu
[5] Unbabel/wmt22-comet-da

ode	Language	Code	Language	Code	Language
afr	Afrikaans	als	Albanian	amh	Amharic
asm	Assamese	bel	Belarusian	ben	Bengali
bos	Bosnian	bul	Bulgarian	cat	Catalan
cym	Welsh	ekk	Estonian	ell	Greek
epo	Esperanto	eus	Basque	fil	Filipino
gle	Irish	glg	Galician	guj	Gujarati
hau	Hausa	heb	Hebrew	hrv	Croatian
hye	Armenian	ind	Indonesian	jav	Javanese
kan	Kannada	kat	Georgian	kaz	Kazakh
khk	Mongolian	khm	Khmer	kir	Kyrgyz
lao	Lao	lit	Lithuanian	lvs	Latvian
mal	Malayalam	mar	Marathi	mkd	Macedonian
mya	Burmese	npi	Nepali	pan	Punjabi
pbt	Pashto	plt	Malagasy	san	Sanskrit
sin	Sinhala	slk	Slovak	slv	Slovenian
som	Somali	srp	Serbian	sun	Sundanese
swh	Swahili	tam	Tamil	tel	Telugu
tha	Thai	tur	Turkish	uig	Uyghur
urd	Urdu	vie	Vietnamese	xho	Xhosa
ydd	Yiddish	zsm	Malay		

Table 15: The 59 unseen languages $\mathcal{L}_{\text{unseen}}$ used for En↔X evaluation after filtering FLORES+ by (i) post-training language coverage of evaluated backbones and (ii) COMET/COMETKIWI language support.

- **CometKiwi**[6] (Rei et al., 2022b), a reference-free version of COMET, useful when reference quality is poor or unavailable.

MQM Evaluation Protocol. To evaluate translation quality across domains and systems, we adopt an enhanced MQM hierarchy tailored to the characteristics of LLMs. Based on the official MQM taxonomy, we introduce additional error types frequently observed in LLM outputs—such as hallucination, semantic repetition, and cross-domain confusion. Our final schema spans seven dimensions: Accuracy, Fluency, Style, Terminology, Others, Source Error, and Non-translation Error, covering 25 fine-grained error categories (Table 16).

For consistent and scalable annotation, we employ DeepSeek-V3 as the automatic scoring model across all evaluations. It is applied uniformly to both traditional and LRMs to ensure fair comparison. The model follows a structured prompt designed to mimic human assessment while enforcing strict formatting and error attribution rules. The full prompt is shown in Figure 3.

B.7 API and Implementation Details

The OpenAI, DeepSeek, and Gemini models used in this study are accessed via the following APIs: gpt-4o-2024-11-20, o1-2024-12-17, o3-mini-2025-01-31, and gpt-5-2025-08-07 for OpenAI; deepseek-chat-2024-12-26 and deepseek-reasoner for DeepSeek; and gemini-2.0-flash and gemini-2.0-flash-thinking-exp-2025-01-21 for Gemini.

[6] Unbabel/wmt22-cometkiwi-da

C Prompt Templates

This section presents the prompts used in our experiments, including those for MQM-style evaluation (Figure 3), difficulty assessment (Figure 4), cold-start SFT data generation (Figure 5), and RL training (Figure 6).

D Case Study: Resolving the Credit Assignment Bottleneck

To elucidate how PAMT resolves the credit assignment bottleneck in reasoning-augmented MT, we contrast two diverse reasoning trajectories sampled from the exact same source sentence during the RL rollout phase (Figures 7 and 8). A persistent challenge in applying LRM to MDMT is terminological rejection—instances where a model successfully retrieves a correct domain constraint in its thought process but fails to execute it in the final translation. Standard sequence-level outcome rewards (e.g., metric-based reward) struggle to optimize this, as they cannot isolate where the reasoning went wrong.PAMT addresses this through dense, step-level supervision (Δ). In the well-aligned trajectory (Figure 7), the policy accurately hypothesizes and commits to the target-domain term ("transitional period") early in the reasoning chain (Step 2). The process reward immediately isolates and validates this pivotal decision with a substantial gain ($\Delta = +3.4018$), explicitly crediting the exact moment of domain alignment. Conversely, the misaligned trajectory (Figure 8) exposes a critical reasoning flaw. In Step 6, the model explicitly contemplates the ground-truth term but actively talks itself out of it, dismissing it in favor of a sub-optimal alternative ("transition period"). While an outcome-based reward would merely assign a slightly lower overall score to the final translation, PAMT's process potential explicitly isolates this incorrect decision, applying a direct step-level penalty ($\Delta = -0.8227$). Takeaway: This comparison confirms that PAMT does not merely optimize final string matching via trial and error. By directly evaluating the marginal process gain (Δ) of each intermediate step, PAMT explicitly steers the model's translation decision, ensuring that domain-faithful reasoning is reliably credited, sustained, and executed.

Category	Error Type	Description
Accuracy	Mistranslation	Inaccurate translation causing semantic distortion.
	Addition	Adding extra information or emotions not in the source.
	Under-translation	Failure to fully convey cultural or contextual nuances.
	Omission	Unintentional exclusion of content from the source text.
	Untranslated	Retaining source text without translation.
	Hallucination	Generating content unrelated to the source text.
	Off-target Translation	Translation misalignment caused by ambiguous input.
	Contradiction	Contradicting itself or the source text.
Fluency	Grammar	Errors in sentence structure or syntax.
	Punctuation	Incorrect use of punctuation marks.
	Spelling	Misspelling of words.
	Semantic Repetition	Unnecessary repetition of words or phrases.
	Logical Incoherence	Lack of logical flow or coherence in translation.
Style	Awkward Expression	Stilted or unnatural phrasing in the target language.
	Unidiomatic Usage	Literal translation causes unnatural wording or grammar.
	Style Inconsistency	Inconsistent stylistic choices within the translation.
	Over-localization	Excessive cultural adaptation leading to distortion.
Terminology	Terminology Inconsistency	Inconsistent translation of the same term.
	Terminology Misuse	Use of incorrect or inappropriate terms for the domain.
	Cross-domain Confusion	Misuse of terms due to domain shifts.
	Incorrect Unit Conversion	Errors in unit conversion (e.g., metric to imperial).
	Formatting	Errors in domain-specific formats (e.g., legal, medical).
Others		Any other errors not covered in the above categories.
Source Error		Errors present in the Source text itself.
Non-translation Error		Translation is unassessable and unrelated to the Source.

Table 16: MQM Hierarchy.

E From GRPO to Process-Aware Credit Assignment

For clarity, we first rewrite vanilla GRPO as terminal-only token credit assignment, and then show how process reward changes the return-to-go at each token position. For brevity, define the sequence-level reward

$$s_i \equiv r_i^{\text{fmt}} + r_i^{\text{out}}. \tag{12}$$

Vanilla GRPO. In vanilla GRPO, the sequence-level reward is placed only on the last valid token:

$$r_{i,t} = \mathbf{1}[t = t_i^{\text{last}}]\, s_i, \qquad (\lambda = 0). \tag{13}$$

The return-to-go is therefore

$$R_{i,t} = \sum_{u=t}^{T_i} r_{i,u} = s_i, \qquad t \le t_i^{\text{last}}. \tag{14}$$

Thus, reward-to-go propagates the same terminal reward to all previous tokens. Let

$$S_i \equiv R_{i,1} = s_i \tag{15}$$

denote the trajectory return. For the G rollouts sampled from the same source sentence, we compute

$$\mu_g = \frac{1}{G}\sum_{i=1}^{G} S_i, \tag{16}$$

$$\sigma_g = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(S_i - \mu_g)^2}. \tag{17}$$

Hence every valid token in the same rollout shares the same advantage:

$$A_{i,t} = \frac{R_{i,t} - \mu_g}{\sigma_g + \epsilon} = \frac{s_i - \mu_g}{\sigma_g + \epsilon}. \tag{18}$$

Algorithm 1: RL training of PAMT

Input : policy π_θ; frozen reference model π_{ref}; training set $\mathcal{D}$; metric set $\mathcal{Q}$; group size G; process weight λ; clip ratio ϵ; KL coefficient β; number of GRPO optimization epochs E_{opt}

Output : trained policy π_θ

while *not converged* **do**
- $\theta_{\text{old}} \leftarrow \theta$
- Sample a prompt minibatch $\mathcal{B} = \{(x_i, y_i^*)\}_{i=1}^{B}$ from $\mathcal{D}$
- Initialize rollout buffer $\mathcal{R} \leftarrow \emptyset$
- `// Phase 1: collect rollouts and compute rewards/advantages with fixed` $\pi_{\theta_{\text{old}}}$
- **for** $i \leftarrow 1$ **to** B **do**
  - Sample G rollouts $\{o_{i,g}\}_{g=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x_i)$
  - **for** $g \leftarrow 1$ **to** G **do**
    - Parse $o_{i,g}$ into `<think>`$z_{i,g}$`</think><answer>`$y_{i,g}$`</answer>` according to Eq. (1)
    - Split $z_{i,g}$ by double newlines into steps $[z_{i,g}^{(1)}, \ldots, z_{i,g}^{(K_{i,g})}]$
    - Cache old token log-probabilities $\ell_{i,g,t}^{\text{old}} \leftarrow \log \pi_{\theta_{\text{old}}}(o_{i,g,t} \mid x_i, o_{i,g,<t})$ for all response token positions t
    - Compute format reward $r_{i,g}^{\text{fmt}}$ and outcome reward $r_{i,g}^{\text{out}}$ using Eq. (2)
    - Initialize $r_{i,g,t}^{\text{proc}} \leftarrow 0$ for all response token positions t
    - Compute the initial process potential $\phi_{i,g,0}$ from the empty reasoning prefix
    - **for** $k \leftarrow 1$ **to** $K_{i,g}$ **do**
      - Construct the prefix context $c_{i,g,k}$ by Eq. (3)
      - Compute the process potential $\phi_{i,g,k}$ by Eq. (4)
      - Compute the step-level process gain $r_{i,g,k}^{\text{proc}} \leftarrow \phi_{i,g,k} - \phi_{i,g,k-1}$ by Eq. (5)
      - Distribute $r_{i,g,k}^{\text{proc}}$ uniformly to all tokens in step $z_{i,g}^{(k)}$ to obtain $r_{i,g,t}^{\text{proc}}$ by Eq. (6)
    - Compute the token-level reward sequence $\{r_{i,g,t}\}$ by Eq. (7)
    - Compute return-to-go $\{R_{i,g,t}\}$ and trajectory return $R_{i,g}^{\text{traj}}$ by Eq. (8)
  - Compute group statistics μ_i, σ_i from $\{R_{i,g}^{\text{traj}}\}_{g=1}^{G}$
  - **for** $g \leftarrow 1$ **to** G **do**
    - **foreach** *response token position* t *in rollout* $o_{i,g}$ **do**
      - Compute the token-level advantage
      $$A_{i,g,t} \leftarrow \frac{R_{i,g,t} - \mu_i}{\sigma_i + \varepsilon}$$
      according to Eq. (9)
    - Add $(x_i, o_{i,g}, \{\ell_{i,g,t}^{\text{old}}\}_t, \{A_{i,g,t}\}_t)$ to rollout buffer $\mathcal{R}$
- `// Phase 2: optimize` π_θ `on the fixed rollout buffer`
- **for** $e \leftarrow 1$ **to** E_{opt} **do**
  - **foreach** *optimization minibatch* $\widetilde{\mathcal{B}} \subset \mathcal{R}$ **do**
    - Compute the importance ratio in Eq. (11) using π_θ and cached $\{\ell_{i,g,t}^{\text{old}}\}$
    - Compute the GRPO objective on $\widetilde{\mathcal{B}}$ by Eq. (10)
    - Take one optimizer step on θ

Step-level process reward. To obtain process-level supervision, we first define a zero-step prefix:

$$c_{i,0} = [x_i, \texttt{<think></think>}, \texttt{<answer>}]. \quad (19)$$

For $k = 1, \ldots, K_i$, let

$$c_{i,k} = [x_i, \texttt{<think>}, z_{i,\le k}, \texttt{</think>}, \texttt{<answer>}]. \quad (20)$$

We define the process potential of the k-step prefix as

$$\phi_{i,k} = \sum_{m=1}^{|y_i^*|} \log \pi_{\text{ref}}\big(y_{i,m}^* \mid c_{i,k}, y_{i,<m}^*\big). \quad (21)$$

The step-level process gain is then

$$\delta_{i,k} \equiv \phi_{i,k} - \phi_{i,k-1}, \qquad k = 1, \ldots, K_i. \quad (22)$$

A positive $\delta_{i,k}$ means that adding step $z_i^{(k)}$ makes the reference translation more predictable under π_{ref}.

To optimize at the token level, we distribute each step gain uniformly over the tokens in that step:

$$g_{i,t} = \begin{cases} \dfrac{\delta_{i,k}}{|z_i^{(k)}|}, & \text{if token } t \text{ belongs to step } z_i^{(k)}, \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

This preserves the total gain of each reasoning step while avoiding a length bias toward longer steps.

Unified token reward. After adding process reward, the token-level reward becomes

$$r_{i,t} = \mathbf{1}[t = t_i^{\text{last}}]\, s_i + \lambda\, g_{i,t}. \tag{24}$$

Compared with vanilla GRPO, the green term is unchanged, while the red term is the newly introduced dense process supervision.

The corresponding return-to-go is

$$\begin{aligned} R_{i,t} &= \sum_{u=t}^{T_i} r_{i,u} \\ &= s_i + \lambda \sum_{u=t}^{T_i} g_{i,u}. \end{aligned} \tag{25}$$

Equation (25) reduces to Eq. (14) when $\lambda = 0$, i.e., standard GRPO.

More explicitly, suppose token t is the ℓ-th token in reasoning step $z_i^{(k)}$, where $\ell \in \{1, \ldots, |z_i^{(k)}|\}$. Then

$$\begin{aligned} R_{i,t} = s_i + \lambda \Bigg(& \frac{|z_i^{(k)}| - \ell + 1}{|z_i^{(k)}|} \delta_{i,k} \\ & + \sum_{j=k+1}^{K_i} \delta_{i,j} \Bigg). \end{aligned} \tag{26}$$

Hence, different reasoning tokens aggregate different suffixes of future process reward. Tokens whose remaining reasoning suffix is more helpful receive larger returns, while tokens followed by harmful steps receive smaller returns. In contrast, for tokens in the <answer> span, all process rewards lie in the past, so

$$R_{i,t} = s_i, \qquad t \in \texttt{<answer>}\ \text{span}. \tag{27}$$

The trajectory return now becomes

$$S_i = R_{i,1} = s_i + \lambda \sum_{k=1}^{K_i} \delta_{i,k}. \tag{28}$$

Advantage normalization. As in GRPO, we normalize the token return using group statistics computed from the updated trajectory returns:

$$\mu_g = \frac{1}{G} \sum_{i=1}^{G} S_i, \tag{29}$$

$$\sigma_g = \sqrt{\frac{1}{G} \sum_{i=1}^{G} (S_i - \mu_g)^2}. \tag{30}$$

The token-level advantage is then

$$A_{i,t} = \frac{R_{i,t} - \mu_g}{\sigma_g + \epsilon}. \tag{31}$$

Unlike vanilla GRPO, $A_{i,t}$ now depends on the token position through the remaining future process reward.

Optimization. The importance ratio is

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid x_i, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid x_i, o_{i,<t})}. \tag{32}$$

We further define the clipped ratio

$$\bar{\rho}_{i,t}(\theta) = \text{clip}\big(\rho_{i,t}(\theta),\, 1-\epsilon,\, 1+\epsilon\big). \tag{33}$$

and the clipped token objective

$$\ell_{i,t}(\theta) = \min\Big(\rho_{i,t}(\theta) A_{i,t}, \bar{\rho}_{i,t}(\theta) A_{i,t}\Big). \tag{34}$$

The GRPO-style clipped loss is

$$\mathcal{L}_{\text{clip}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} \ell_{i,t}(\theta), \tag{35}$$

and the full objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{clip}}(\theta) + \beta\, \text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big). \tag{36}$$

In this view, vanilla GRPO is simply the special case $\lambda = 0$, where reward-to-go propagates the same terminal reward to all previous tokens. PAMT keeps the same GRPO optimization backbone, but changes the token return by adding dense process rewards, making token credit assignment position-dependent.

Toy example. Consider a rollout with three reasoning steps. Suppose their step-level gains are

$$(\delta_{i,1},\, \delta_{i,2},\, \delta_{i,3}) = (0.4,\, -0.3,\, 0.2),$$

and their lengths are

$$(|z_i^{(1)}|,\, |z_i^{(2)}|,\, |z_i^{(3)}|) = (2,\, 1,\, 1).$$

Intuitively, the first step is helpful, the second step introduces process drift, and the third step partially corrects it.

By distributing each step gain uniformly over its tokens, the token-level process rewards become

$$(g_{i,1},\, g_{i,2},\, g_{i,3},\, g_{i,4}) = (0.2,\, 0.2,\, -0.3,\, 0.2).$$

Suppose the sequence-level reward is $s_i = 0.8$ and $\lambda = 1$. Then the return-to-go on the four reasoning tokens is

$$(R_{i,1},\, R_{i,2},\, R_{i,3},\, R_{i,4}) = (1.1,\, 0.9,\, 0.7,\, 1.0),$$

because each token receives the sequence-level reward s_i plus the suffix sum of the remaining process rewards. For example,

$$R_{i,2} = 0.8 + (0.2 - 0.3 + 0.2) = 0.9,$$

while

$$R_{i,3} = 0.8 + (-0.3 + 0.2) = 0.7.$$

For tokens in the <answer> span, all process rewards already lie in the past, so

$$R_{i,t} = 0.8.$$

Under vanilla GRPO, all of these positions would instead receive the same return 0.8. This example shows that process-aware credit assignment is not simply "earlier is better"; rather, a token receives a higher return when the remaining reasoning suffix is more helpful for producing the reference translation.

You are a professional translation quality evaluator following the MQM (Multidimensional Quality Metrics) framework.

Task Instructions:

1. Compare the Prediction against both the Source and the Reference.
2. Identify up to five of the most serious Errors for each translation sentence, using the MQM error types listed below.
3. Assign exactly one severity level to each error.
4. Special handling for two specific error types:
   - **Source Error**: Errors present in the Source text itself.
   - **Non-translation Error**: Translation is unassessable and unrelated to the Source.
5. If no errors are found, return an empty JSON list [].
6. Only output a valid JSON object. Do not include any additional text, comments, or explanations.

MQM Error Types (See Table 16 for the complete hierarchy):

- Mistranslation: Incorrect translation that alters or distorts meaning.
- Addition: Insertion of information or emotion not present in the source.
- Under-translation: Partial omission of relevant or necessary information.
- Omission: Complete exclusion of content present in the source.
- Untranslated: Source text is copied without being translated.

Severity Levels:

- Minor: Slight impact on readability or style; meaning remains clear.
- Major: Significant impact on usability, comprehension, or meaning.

Output Format:

```
{
  "errors": [
    {
      "error_type": "Mistranslation",
      "severity": "Major",
      "explanation": "The correct translation should be ... "
    }
  ]
}
```

Source: `{source_text}`
Reference: `{reference_text}`
Prediction: `{prediction_text}`

Figure 3: Full prompt used to calculate MQM scores with DeepSeek-V3.

Your task is to assess the difficulty of translating a given {src_lang} sentence into {tgt_lang}. Please evaluate the difficulty based on the following criteria and output the result in JSON format, with the key "level":

1. Sentence complexity: Determine if the sentence is a simple sentence, a compound sentence, or includes subordinate clauses and other complex structures.
2. Vocabulary difficulty: Assess whether the sentence contains commonly used words or specialized terms or slang.
3. Grammar differences: Analyze if the sentence's grammatical structure is similar to or differs significantly from {tgt_lang}.
4. Contextual understanding: Consider whether understanding specific cultural contexts or background knowledge is necessary for accurate translation.

Provide a difficulty level (1-5), with 1 being the easiest and 5 being the most difficult.

And output the difficulty level in the following JSON format:

```
{
    "level": "difficulty level"
}
```

Here is the {src_lang} sentence: {src_text}

Figure 4: Full prompt used for evaluating translation difficulty with DeepSeek-V3.

Translate the following {src_lang} text into {tgt_lang} while maintaining the domain style of the source text.

Source: {src_text}

Figure 5: Full prompt used for cold start SFT.

Translate the following {src_lang} text into {tgt_lang}.

Source: {src_text}

Figure 6: Full prompt used for RL stage.

Case Study: Trajectory of Process Potential in PAMT (Law Domain)

Language Pair: German → English (de-en) **Scores:** Format: 1.0000 | Answer: 2.4624 | Total: 3.4624

Source Input:
- "Übergangszeitraum": der Zeitraum zwischen dem 1. Januar 1999 und dem 31, Dezember 2001.

Reference:
- "transitional period" shall mean the period between 1 January 1999 and 31 December 2001.

Prediction:
- "Transitional period": the period between 1 January 1999 and 31 December 2001.

Step-by-Step Reasoning Analysis: *Initial* $\Phi = -30.7478$

- **Step 1:** Okay, let's tackle this translation. The original German sentence is about a term: "Übergangszeitraum": der Zeitraum zwischen dem 1. Januar 1999 und dem 31. Dezember 2001.
  Metrics: $\Phi = -35.9806$ $\Delta = -5.2327$
- **Step 2:** First, translating the term "Übergangszeitraum" as "transitional period" makes sense in many contexts. "Transitional" is straightforward here. The phrase "der Zeitraum zwischen dem 1. Januar 1999 und dem 31. Dezember 2001" becomes "the period between 1 January 1999 and 31 December 2001."
  Metrics: $\Phi = -32.5788$ $\Delta = +3.4018$
- **Step 3:** I should check the date format. In European-style writing, they use the ordinal, like "31. December," but in English, we use numerals without the ordinal suffix, so "31 December." Also, the years are written as 1999 and 2001.
  Metrics: $\Phi = -32.9279$ $\Delta = -0.3491$
- **Step 4:** Putting it all together: "- 'Transitional period': the period between 1 January 1999 and 31 December 2001."
  Metrics: $\Phi = -30.3991$ $\Delta = +2.5288$
- **Step 5:** I need to confirm if there's any other terminology here. The style is formal, so "period" is correct. No need for extra words. The original uses "der Zeitraum," which is straightforward, so "the period" works.
  Metrics: $\Phi = -30.0263$ $\Delta = +0.3728$
- **Step 6:** I don't see any technical jargon beyond this, so the translation should be straightforward. No need for adjustments.
  Metrics: $\Phi = -30.7830$ $\Delta = -0.7567$
- **Step 7:** Also, notice that in the original, there's a hyphen at the start, which I should keep. So the translation starts with "-".
  Metrics: $\Phi = -30.4701$ $\Delta = +0.3129$
- **Step 8:** Double-checking the dates: January 1 to December 31. Yes, that's the correct range.
  Metrics: $\Phi = -30.4014$ $\Delta = +0.0687$
- **Step 9:** So the final translation should be:
  Metrics: $\Phi = -30.7416$ $\Delta = -0.3401$
- **Step 10:** - "Transitional period": the period between 1 January 1999 and 31 December 2001.
  Metrics: $\Phi = -29.0100$ $\Delta = +1.7316$
- **Step 11:** Yes, that's concise and matches the formal style required. The colon is kept as in the original.
  Metrics: $\Phi = -28.6008$ $\Delta = +0.4091$
- **Step 12:** I think that's all. No other nuances in the German sentence. The translation should be accurate and precise, which this seems to be.
  Metrics: $\Phi = -28.9584$ $\Delta = -0.3576$
- **Step 13:** I don't think there's any idiomatic language here, so a direct translation works. No need for adjustments to the structure.
  Metrics: $\Phi = -28.8792$ $\Delta = +0.0792$
- **Step 14:** So the final translation is correct, and the style is formal and concise, as required.
  Metrics: $\Phi = -27.7316$ $\Delta = +1.1476$

Figure 7: A detailed case study demonstrating the trajectory of Process Potential (Φ) and step-level reward gain (Δ) during intermediate translation reasoning. Positive Δ values (green) indicate steps that increase the likelihood of the reference translation.

Case Study: Trajectory of Process Potential (Terminological Rejection)

Language Pair: German → English (de-en) **Scores:** Format: 1.0000 | Answer: 2.0548 | Total: 3.0548

Source Input:
- "Übergangszeitraum": der Zeitraum zwischen dem 1. Januar 1999 und dem 31, Dezember 2001.

Ground Truth:
- "transitional period" shall mean the period between 1 January 1999 and 31 December 2001.

Final Answer:
- "Transitional period": the period from 1 January 1999 to 31 December 2001.

Step-by-Step Reasoning Analysis: *Initial* $\Phi = -30.7478$

- **Step 1:** Okay, I need to translate the German sentence "- 'Übergangszeitraum': der Zeitraum zwischen dem 1. Januar 1999 und dem 31, Dezember 2001." into English. First, let's break down the sentence.
  Metrics: $\Phi = -35.0311$ $\Delta = -4.2833$
- **Step 2:** "Übergangszeitraum" is "transition period" or "transitional period". "den Zeitraum zwischen dem 1. Januar 1999 und dem 31. Dezember 2001" translates to "the period from 1 January 1999 to 31 December 2001".
  Metrics: $\Phi = -34.4548$ $\Delta = +0.5764$
- **Step 3:** So the translation would be: "- 'Transition period': the period from 1 January 1999 to 31 December 2001."
  Metrics: $\Phi = -35.7834$ $\Delta = -1.3287$
- **Step 4:** Now, I need to identify the domain. The text is about a specific time frame, so it's likely legal or governmental. The style should be formal, so using "transition period" with a colon after the term is correct.
  Metrics: $\Phi = -36.7196$ $\Delta = -0.9362$
- **Step 5:** Check for any nuances. The German uses "bzw." (between), so the translation is accurate. Also, the dates are straightforward, no need for additional words.
  Metrics: $\Phi = -36.7160$ $\Delta = +0.0036$
- **Step 6:** I should confirm that "Übergangszeitraum" is best as "transition period" rather than "transitory period" or "transitional period". Since the original uses "Übergangszeitraum", "transition" fits well.
  Metrics: $\Phi = -37.5387$ $\Delta = -0.8227$
- **Step 7:** No other adjustments needed. The sentence structure is simple, so the translation should mirror that simplicity.
  Metrics: $\Phi = -36.3380$ $\Delta = +1.2007$
- **Step 8:** Final translation: "- 'Transition period': the period from 1 January 1999 to 31 December 2001."
  Metrics: $\Phi = -36.3168$ $\Delta = +0.0212$
- **Step 9:** Yes, that's concise and formal, which suits the domain style. No extra fluff, just the necessary information.
  Metrics: $\Phi = -35.6590$ $\Delta = +0.6578$
- **Step 10:** I think that's all. The domain is probably legal or financial, but since the text is a definition, the style is formal. The translation seems to meet that requirement.
  Metrics: $\Phi = -35.1589$ $\Delta = +0.5001$
- **Step 11:** I don't see any other issues here. The numbers and dates are correctly rendered, and the term is appropriately translated without any unnecessary complexity.
  Metrics: $\Phi = -34.6357$ $\Delta = +0.5232$
- **Step 12:** So, the final answer is correct and the domain identification is accurate. The translation reflects the formal style required by the context.
  Metrics: $\Phi = -34.4831$ $\Delta = +0.1526$

Figure 8: Case study demonstrating terminological rejection. In Step 6 ($\Delta = -0.8227$), the model explicitly considers the correct ground-truth terminology ("transitional period") but erroneously talks itself out of it, dismissing it in favor of "transition period". The process potential accurately penalizes this active deviation from the reference terminology.