Enhancing Machine Translation with Self-Supervised Preference Data

Haoxiang Sun[1*], **Ruizhe Gao**[2], **Pei Zhang**[2], **Baosong Yang**[2†], **Rui Wang** [1†]
[1]Shanghai Jiao Tong University
[2]Tongyi Lab, Alibaba Group
[1]{sunny_sjtu,wangrui12}@sjtu.edu.cn
[2]{gaoruize.grz, xiaoyi.zp,yangbaosong.ybs}@alibaba-inc.com

Abstract

Model alignment methods like Direct Preference Optimization (Rafailov et al., 2024) and Contrastive Preference Optimization (Xu et al., 2024b) have enhanced machine translation performance by leveraging preference data to enable models to reject suboptimal outputs. During preference data construction, previous approaches primarily rely on humans, strong models like GPT4 (OpenAI, 2023) or model self-sampling. In this study, we first explain the shortcomings of this practice. Then, we propose **Self-Supervised Preference Optimization (SSPO)**, a novel framework which efficiently constructs translation preference data for iterative DPO training. Applying SSPO to 14B parameters large language models (LLMs) achieves comparable or better performance than GPT-4o on FLORES and multi-domain test datasets. We release an augmented MQM dataset in https://github.com/sunny-sjtu/MQM-aug.

1 Introduction

Enhancing the capabilities of open source large language models (LLMs) (Bai et al., 2023; Touvron et al., 2023; Jiang et al., 2023) in machine translation has been extensively explored in previous research. ALMA (Xu et al., 2024a) and Aya 23 (Aryabumi et al., 2024) reach top-tier performance through continued pre-training on large monolingual corpora and supervised fine-tuning (SFT) on high-quality parallel translation data. While SFT lacks a mechanism to prevent the model from rejecting mistakes in translations (e.g. mistranslation, over-translation), model alignment methods like Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), Direct Preference Optimization (DPO) (Rafailov et al., 2024) and Contrastive Preference Optimization (CPO) (Xu et al., 2024b) further improve machine translation performance by leveraging preference data to enable models to reject suboptimal outputs.

High-quality preference data is crucial for effective model alignment. Current approaches to constructing translation preference data typically rely on human annotations (Xu et al., 2024c; Ramos et al., 2024), model self-sampling (Yang et al., 2024b) or stronger models (Xu et al., 2024b).

This practice faces three major challenges: (1) high cost of querying humans or strong models (2) distributional discrepancy between positive and negative examples from different models, which leads to training instability. (3) insufficient quality contrast between self-sampled positive and negative examples, which weakens reward signals.

To address these challenges, we propose a self-supervised framework for moderate-sized models ($\sim$14B parameters) that constructs high-quality preference data without relying on stronger models or human annotations. **Our key insight is to equip LLMs with three core capabilities: translation generation, error annotation, and error correction.** This allows LLMs to utilize monolingual data by translating, identifying potential errors, and generating corrected versions. The quality gap between the initial and corrected translations naturally forms preference pairs for model alignment. **This method also supports continuous improvement through an iterative refinement process.** After generating error annotations, we further fine-tune the base model with these examples, creating a specialized error detector which becomes increasingly sensitive to common translation mistakes made by the current model. This enhanced detector provides targeted supervision for the DPO-aligned translation model. As the translation model improves via DPO training, the error detector adapts to new error patterns, progressively enhancing overall translation quality.

Our contributions are summarized as follows:

[*]Work done during internship at Tongyi Lab.
[†]Rui Wang and Baosong Yang are co-corresponding authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 23916–23934
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

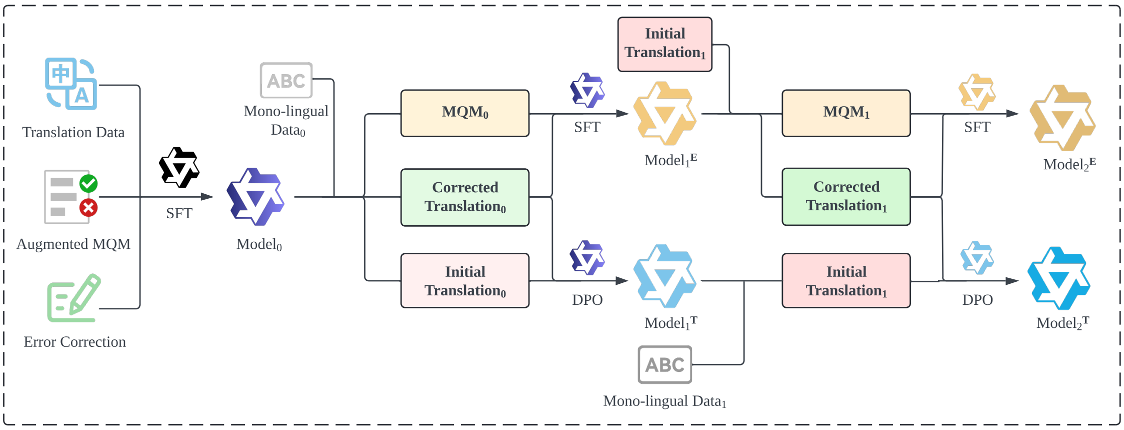

Figure 1: **Overview of SSPO Framework.** The process starts by training Model$_0$ with three types of data: parallel translation pairs, augmented MQM (Multidimensional Quality Metrics) annotations with error explanation and suggested correction, and error correction samples leveraging MQM annotations for corrected translations. The framework then iteratively improves through two paths: (1) SFT with self-generated MQM annotations and corrections to get stronger **Error** detector Model$^{\mathbf{E}}$ (2) DPO with <intial translation, corrected translation> preference pairs to get stronger **Translator** Model$^{\mathbf{T}}$. Each iteration incorporates new monolingual data to expand domain coverage. Deeper model colors indicate enhanced capabilities.

- We propose SSPO, a self-supervised mechanism that enables LLMs to iteratively generate high-quality translation preference data for DPO training. SSPO's effectiveness is validated across multiple languages, domains and models, achieving consistent improvements in translation performance without relying on external human or model annotations.

- We identify a good strategy for composing translation preference data, showing that integrating model-generated preferences with external high-quality data (from human experts or strong models) during DPO training yields superior performance compared to using either source alone.

- We release an augmented MQM annotation dataset to boost LLMs' performance in translation-related tasks.

2 Self-supervised Preference Optimization

SSPO is a paradiam designed to generate high-quality translation preference data for iterative preference optimization. Figure 1 provides an overview of SSPO. We begin by describing the initialization of Model$_0$, the foundation of our framework.

2.1 Initialization of Model$_0$

Training Set. Model$_0$ is initialized using three complementary types of training data: parallel translation data, augmented MQM (Multidimensional Quality Metrics) annotations, and error correction data.

MQM offers a detailed assessment of translation quality by identifying specific error types, spans, and severity levels. For each error marked in the MQM annotations, we prompt GPT-4o* (OpenAI, 2024) for correction suggestions.

The error correction data is generated by prompting GPT-4o with the source text, initial translation, and MQM annotations to provide an improved translation. We show our prompts for training data construction in Appendix A.1.

Supervised fine-tuning with these data equips Model$_0$ with three key capabilities: translation generation, error annotation and error correction.

Design Rationale. We decompose the seemingly continuous chain of error annotation and correction into two separate tasks. This design is motivated by two key observations: (1) generating accurate MQM annotations is the most challenging part, requiring deep understanding of translation errors and improvements. (2) the correction process is straightforward, mainly applying MQM annotations to the initial translation. This separation focuses our iterative refinement on enhancing

*We use `gpt-4o-0806` available from the OpenAI API.

the model's error annotation capabilities, which is more crucial for generating high-quality preference data.

2.2 Self-Supervised Preference Data Construction

Overview. As depicted in Figure 2, Model_0 processes monolingual input through three sequential steps:

1. **Translation Generation.** For source text $\mathbf{x} \sim \mathcal{D}_i$, Model_0 generates an initial translation $\mathbf{y} = \text{Model}_0(\mathbf{x})$. $\mathcal{D}_i$ denotes the distribution of monolingual texts in iteration i.

2. **Error Annotation.** The model then performs error analysis by generating MQM-style annotations $\{\mathbf{e}_1, ..., \mathbf{e}_n\} = \text{Model}_0(\mathbf{x}, \mathbf{y})$. If no errors are detected, it outputs "There is no error in the translation." and skips the correction step. Otherwise, each identified error $\mathbf{e}_i$ is detailed as a tuple $\{\mathbf{loc}_i, \mathbf{sev}_i, \mathbf{exp}_i, \mathbf{sugg}_i\}$ compromising:

   - $\mathbf{loc}_i$: Erroneous text span.
   - $\mathbf{sev}_i$: Error severity (major/minor).
   - $\mathbf{exp}_i$: Explanation for the error.
   - $\mathbf{sugg}_i$: Suggested improvement.

3. **Error Correction.** For translations containing errors, the model generates an improved version: $\mathbf{y}' = \text{Model}_0(\mathbf{x}, \mathbf{y}, \{\mathbf{e}_1, ..., \mathbf{e}_n\})$.

This process yields preference pairs $\langle \mathbf{y}, \mathbf{y}' \rangle$ for preference optimization.

While these three capabilities are initially unified in Model_0, they are later separated into two specialized models: $\text{Model}^{\mathbf{T}}$ for **Translation** and $\text{Model}^{\mathbf{E}}$ for **Error** annotation and correction. Specifically, for $Iter_i (i > 0)$ and input $\mathbf{x} \sim \mathcal{D}_i$, Model_i^T generates the initial translation $\mathbf{y} = \text{Model}_i^T(\mathbf{x})$. Then Model_i^E identified potential errors $\{\mathbf{e}_1, ..., \mathbf{e}_n\} = \text{Model}_i^E(\mathbf{x}, \mathbf{y})$ and produces an improved version $\mathbf{y}' = \text{Model}_i^E(\mathbf{x}, \mathbf{y}, \{\mathbf{e}_1, ..., \mathbf{e}_n\})$ if errors are detected, forming a preference pair $\langle \mathbf{y}, \mathbf{y}' \rangle$.

Automatic Filtering & Domain Expansion. A key advantage of our framework is its ability to utilize diverse monolingual data across different domains. In each iteration, we introduce new monolingual texts from varied domains, where Model^T generates translations and Model^E automatically screens for errors. Only translations with identified issues undergo improvement, creating selected and high-quality preference pairs. This approach enables continuous domain knowledge expansion while ensuring efficient preference data generation through automatic error filtering.

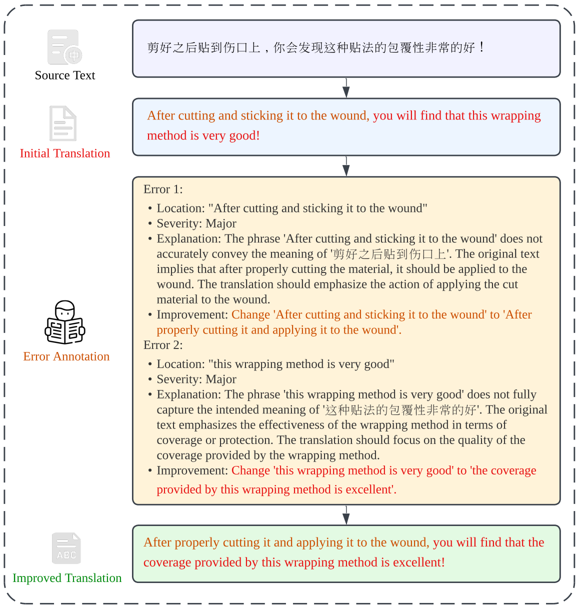

Figure 2: Self-Supervised Preference Data Construction

2.3 Preference Optimization

Following the construction of preference pairs, the next step is to optimize the translation model using these self-supervised preferences. Our method is compatible with various preference optimization techniques, such as DPO (Rafailov et al., 2024), CPO (Xu et al., 2024b) and SimPO (Meng et al., 2024). We choose DPO for its training stability. Given our self-supervised preference dataset P_i containing tuples of $(\mathbf{x}, y_w, y_l)$, where y_w and y_l are the better and worse translations, respectively, the final training objective integrates DPO loss with SFT loss:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{DPO}} + \alpha \cdot \mathcal{L}_{\text{SFT}}, \tag{1}$$

where the DPO loss and SFT loss are defined as:

$$\begin{aligned} \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -E_{(\mathbf{x}, y_w, y_l) \sim P_i}[\log \sigma \\ (\beta \log \frac{\pi_\theta(y_w \mid \mathbf{x})}{\pi_{\text{ref}}(y_w \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(y_l \mid \mathbf{x})}{\pi_{\text{ref}}(y_l \mid \mathbf{x})})], \end{aligned} \tag{2}$$

$$\mathcal{L}_{\text{SFT}}(\pi_\theta) = -E_{(\mathbf{x}, y_w) \sim P_i}[\log \pi_\theta(y_w \mid \mathbf{x})]. \tag{3}$$

Here, π_θ denotes the translation model Model_i^T being trained in the current iteration, π_{ref} is the reference model Model_{i-1}^T from the previous iteration. The parameter β is a temperature parameter controlling the sharpness of the preference learning, and α_{sft} adjusts the weight of SFT loss for training stability. In our preference pairs, y_w is the improved translation $\mathbf{y}'$ generated by Model_{i-1}^E, and y_l is the original translation $\mathbf{y}$ from Model_{i-1}^T.

2.4 Self-Training for Error Annotation

To maintain effective quality assessment as the translation model improves through DPO, we enhance Model^E via self-training. The training data D_{SFT} comprises three components: (1) error annotation pairs $(\mathbf{x}, \mathbf{y}) \rightarrow \{\mathbf{e}_1, ..., \mathbf{e}_n\}$, (2) error correction pairs $(\mathbf{x}, \mathbf{y}, \{\mathbf{e}_1, ..., \mathbf{e}_n\}) \rightarrow \mathbf{y}'$, (3) error-free pairs $(\mathbf{x}, \mathbf{y}) \rightarrow$ "No error". The last component prevents over-criticism by helping the model recognize high-quality translations.

This self-training process enables Model^E to maintain a balanced capability in identifying genuine translation errors and recognizing high-quality translations, ensuring effective quality assessment for the enhanced translation model.

3 Experiment

We carry out comprehensive experiments to demonstrate the effectiveness of SSPO in enhancing LLMs' machine translation performance.

3.1 Data

We consider 10 translation directions in the paper: da↔en, de↔en, fr↔en, id↔en, zh↔en. As illustrated in Section 2.1, our training dataset consists of three complementary components, detailed data statistics can be found in Appendix A.2-A.4.

Translation Data. For en→de,fr→en,zh↔en, we collect high-quality parallel translation pairs from WMT News Task development and test sets across multiple years. For other language pairs, we sample from News Commentary v18.1 and Europarl v10.

Augmented MQM Annotations. For en→de and zh→en, we collect original MQM-style error annotations from WMT Metrics shared tasks in 2020,2021,2023 (Freitag et al., 2021, 2023). For de→en, fr→en, we collect original MQM-style error annotations from a bio-domain MQM dataset (Zouhar et al., 2024). These data are augmented following the steps in Section 2.1. We provide an example of the augmented MQM annotations in Appendix A.3.

Error Corrections. We construct error correction pairs by sampling from the MQM annotations and manually correcting the identified errors with GPT-4o.

Monolingual Data. We collect monolingual data from open source internet then conduct length and perplexity filtering (detailed in Appendix A.5).

3.2 Models and Training

We apply SSPO to Qwen2.5-14B-Base (Yang et al., 2024a) (14B parameters) and Mistral-Nemo-Base-2407 (Jiang et al., 2023) (12B parameters). The process begins with supervised fine-tuning on our dataset to develop Model_0, which functions as both the initial translation model Model_0^T and the error detection model Model_0^E. Each iteration i involves three steps: (1) Model_i^T generates translations for monolingual inputs, (2) Model_i^E identifies errors and creates preference pairs for DPO optimization of Model_{i+1}^T, (3) Model_i^E undergoes self-training to produce Model_{i+1}^E. We adopt LoRA (Hu et al., 2021) in DPO and SFT traing. Our prompts, training parameters and implementation environments are provided in Appendix B.1.

3.3 Evaluation

Multi-domain Test sets. For zh→en direction, we employ a 10-domain test suite (Table 1) to evaluate cross-domain generalization. Full test set details are in Appendix B.2.

Domain	Count	Domain	Count
Industry	3,487	Finance	1,322
Talk	2,599	E-commerce	1,001
IT	2,293	Thesis	625
News	1,875	Biology	575
Literary	1,514	Science	503

Table 1: Distribution of test samples across different domains for zh→en direction

Multi-lingual Test sets. We evaluate on FLORES-200 test sets (Team et al., 2022) to assess cross-lingual transferability. Full test set details are in Appendix B.3.

Metrics. Following the arguments in CPO Xu et al. (2024b), which demonstrates that human-written references are not always superior to model outputs and advocates for reference-free evaluation, we adopt reference-free metrics for our evaluation. Specifically, we use: (1) `Unbabel/XCOMET-XXL`, refered as **XCOMET** (Guerreiro et al., 2023); (2) `Unbabel/wmt23-cometkiwi-da-xxl`, referred as **KIWI** (Rei et al., 2023); and (3) `google/metricx-23-qe-xxl-v2p0`, referred as **METRICX** (Juraska et al., 2023).

3.4 Baselines

SoTA Models. We compare with state-of-the-art open-source models including Unbabel's **TowerInstruct** (Alves et al., 2024b) and **ALMA-13B-R** (Xu et al., 2024b) (trained with GPT-4 preference data via CPO). For closed-source models, we benchmark against `GPT-4o-0806`.

Pipelines. We evaluate our method against three open-source preference data construction pipelines on multi-domain `zh`→`en` test set. Instead of starting from Model_0, these methods begin with Model-Trans, trained on translation data in our dataset solely for translation generation. (1) **Self-Sampling + XCOMET**: Generate multiple translations through model self-sampling, then select highest and lowest scoring samples based on XCOMET scores; (2) **XCOMET + xTower**: Generate initial translations, identify errors using XCOMET, then refine translations with xTower (Treviso et al., 2024) (Unbabel's 13B correction model trained on GPT-4 annotated data) as positive examples; (3) **XCOMET + Qwen2.5-Plus**: Similar to (2) but using Qwen2.5-Plus with 5-shot prompting for error correction.

3.5 Multi-domain Results

Multi-domain Evaluation. We systematically evaluate cross-domain generalization using Qwen-2.5-14B-Base on `zh` → `en` direction. As shown in Table 2, our iterative optimization achieves consistent gains across all domains. The scores shown are over all domains, with detailed domain-specific results provided in Appendix B.4. These improvements are largely due to the diverse nature of our monolingual data, which enriches the model with domain-specific knowledge.

Comparison with Open-source Pipelines. We conduct comparative experiments to evaluate different preference data construction approaches. Following Section 3.4, we first train Qwen-2.5-14B-Base solely on translation data to obtain Qwen-Trans, which serves as the initial translation model for all pipeline methods. To ensure fair comparison, we use the same set of monolingual data - specifically, the source sentences corresponding to the 12,800 preference data pairs used in training our Qwen-Model_1^T.

Models	Metrics		
	KIWI	XCOMET	METRICX↓
Baseline Models			
Qwen-Trans	79.32	90.28	4.6499
ALMA-13B-R	79.19	91.12	4.5454
TowerInstruct	78.92	90.22	4.7780
GPT-4o-0806	80.29	91.54	4.3310
Pipelines			
Self-sampling + XCOMET	79.20	90.43	4.6686
XCOMET + xTower	79.85	**92.93**	4.2451
XCOMET + Qwen2.5-Plus	80.03	91.64	4.4125
Our Method			
Qwen-Model_0	79.42	90.37	4.6393
Qwen-Model_1^T	80.38	92.40	4.2382
Qwen-Model_2^T	**80.67**	**92.66**	**4.2187**

Table 2: Performance comparison on zh→en multi-domain test sets. ↓ means lower is better.

Table 2 presents the results on reference-free metrics. We summarize two key observations:

Integration of error-related data enhances translation performance. Error annotation and correction training data enhances translation performance (Qwen-Model_0 outperforms Qwen-Trans). This improvement can be attributed to the implicit translation knowledge embedded in error-related data.

SSPO generate high-quality preference data. Through a single iteration of SSPO, Qwen-Model_1^T demonstrates significant improvements compared to Qwen-Model_0 (+0.96 KIWI, +2.03 XCOMET, -0.4 METRICX). Remarkably, it outperforms GPT-4o-0806 across all metrics. In contrast, the self-sampling + DPO approach, which relies on 14B model's self-sampling for preference data, shows limited effectiveness. Error correction pipelines using xCOMET annotations (XCOMET + xTower and XCOMET + Qwen2.5-Plus) also demonstrate great improvements in translation quality, but they show strong bias towards XCOMET metrics.

We conduct additional LLM evaluation using Claude-3.5[*] (Anthropic, 2023) to compare Qwen-Model_1^T with the best-performing open-source pipeline (XCOMET + xTower) through pairwise comparison. To reduce position bias, we per-

[*]We use `claude-3-5-20241022` available from Anthropic API.

form two rounds of evaluations with swapped positions of candidate translations. **Qwen-Model$_1^T$ outperforms xTower Pipeline with a net win rate of 8.5%.** Evaluation details are listed in Appendix B.5.

3.6 Multi-lingual Results

We conduct 2 iterations of self-supervised optimization on Qwen2.5-14B-Base and Mistral-Nemo-Base-2407 across 10 language directions. For zh $\rightarrow$ en direction, we evaluate on our multi-domain test sets, while for other language pairs, we use the FLORES200 testset. Primary results for xx$\rightarrow$en and en$\rightarrow$xx are shown in Figure 3. Following Section 3.3, we report reference-free metrics, full results are detailed in Appendix B.6. Key findings are summarized below.

Progressive Performance Enhancement. Both Qwen and Mistral models show consistent gains across iterations, with Model$_2^T$ outperforming Model$_1^T$ and Model$_1^T$ outperforming Model$_0$ across all language pairs. Notably, for xx$\rightarrow$en direction, both Qwen-Model$_2^T$ and Mistral-Model$_2^T$ achieve comparable or superior performance to GPT-4o-0806. Although a performance gap remains with GPT-4o-0806 in en $\rightarrow$ xx direction, our approach still demonstrates substantial improvements (e.g. +0.62 KIWI, +0.37 XCOMET, -0.0526 METRICX when Mistral-Model$_2^T$ v.s. Mistral-Model$_0$). These consistent gains across different translation directions validate the effectiveness of our iterative optimization approach.

Asymmetric Performance Gains Across Directions. The improvement patterns differ between xx $\rightarrow$ en and en $\rightarrow$ xx translations, primarily due to the distribution of MQM annotation training data. For xx $\rightarrow$ en, abundant MQM data enables high-quality error detection in the first iteration, leading to strong preference data and substantial gains. However, the second iteration shows limited improvement due to a growing capability gap: while Model$_1^T$ achieves significant enhancement through DPO training on high-quality preference data, Model$_1^E$, trained solely on self-generated annotations, shows marginal improvement in error detection. Thus it struggles to identify subtle errors in increasingly better translations. Conversely, for en $\rightarrow$ xx where MQM data is scarce, limited initial error detection capability leads to modest improvements of Model$_1^T$. When Model$_1^E$ is trained on self-generated annotations, its error detection capability improves moderately, still sufficient to identify errors in Model$_1^T$'s translations, thus enabling further performance gains in the second iteration.

Cross-lingual Transfer of MQM Annotation Ability. Despite our training data only containing MQM annotations for de$\leftrightarrow$en, fr$\rightarrow$en, and zh$\rightarrow$en directions, the models successfully generalize to other language pairs, effectively constructing preference data that leads to improved translation performance. This phenomenon is also discovered by Uhlig et al. (2025). We find this generalization ability is influenced by the model's inherent linguistic capabilities: for language pairs where the base model shows strong performance, high-quality preference data can be generated even without corresponding MQM data. This is exemplified by Mistral's significant first-iteration improvements in da$\rightarrow$en translation.

4 Analyses

We conduct extensive analyses to investigate three critical aspects of our approach. First, we explore **how the amount of monolingual data used within a single iteration impacts the optimization results**, aiming to determine the optimal data quantity for maximizing translation quality. Second, we examine **the impact of error correction strategies when generating preference data**, specifically comparing two approaches: correcting all errors (both major and minor) in each iteration versus a progressive strategy that focuses on major errors in the first iteration and addresses all errors in subsequent iterations. Third, we investigate **whether incorporating external high-quality preference data can further enhance translation quality.** We use Qwen2.5-14B-Base and focus on zh $\rightarrow$ en direction. These analyses aim to provide deeper insights into the mechanisms and optimization strategies of our approach. We analyze the evolvement of Model^E in Appendix C and provide an ablation study of our approach in Appendix D.

4.1 Impact of Monolingual Data Amount

In the first iteration of zh$\rightarrow$en translation, we observe that model performance plateaus after training on 12,800 preference data pairs, with additional data yielding diminishing returns. To investigate the optimal data utilization strategy, we sys-

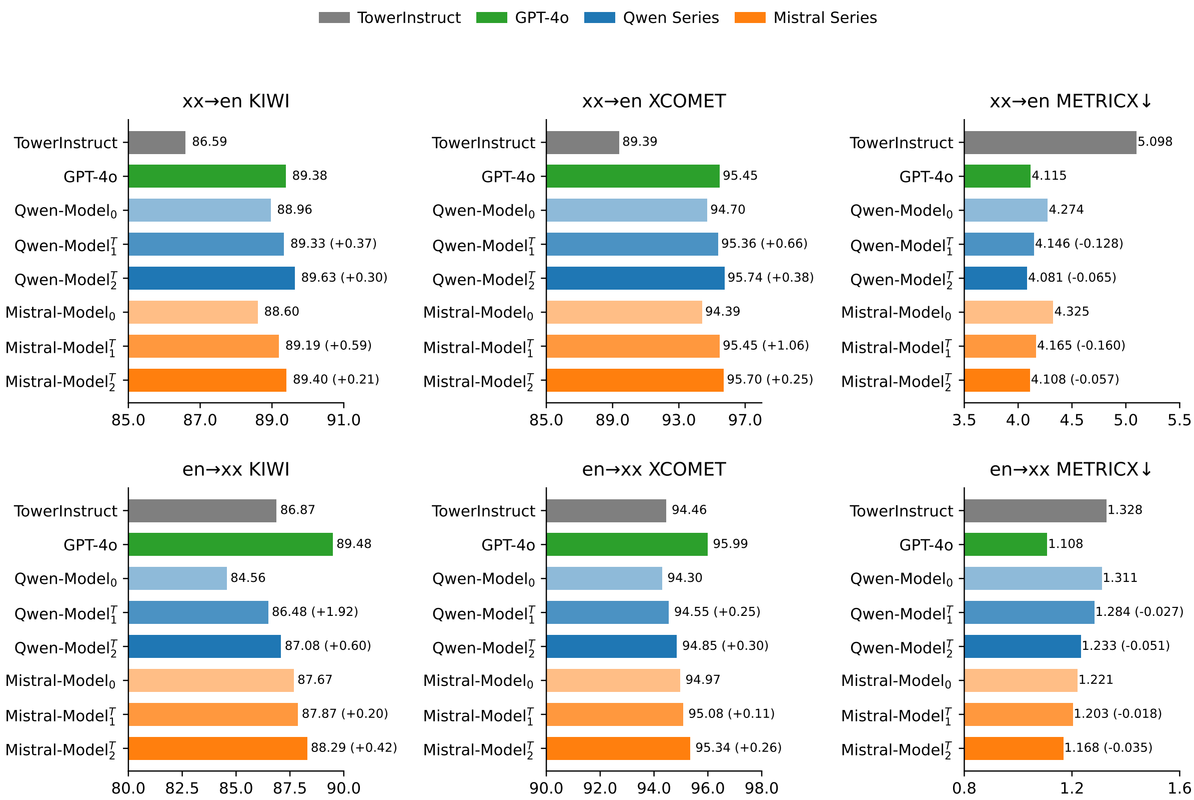

Figure 3: Average results for en-xx and xx-en translation directions.

tematically partition the full dataset into 2, 3, 4 equal subsets for multi-iteration training. The key question we want to address is whether to use all available monolingual data in a single iteration until performance saturates, or to distribute it across multiple iterations for gradual improvement.

Results in Table 3 demonstrate clear superiority of the single-iteration strategy. When we split the 12,800 samples into multiple portions, the cumulative improvement after multiple iterations fails to match the performance achieved by using all data at once. This indicates that using the entire data in a single iteration optimizes performance more effectively than incremental updates with smaller portions.

Models	Metrics		
	KIWI	XCOMET	METRICX↓
Model$_0$	79.42	90.37	4.6393
Iter1-1	**80.38**	**92.40**	**4.2382**
Iter2-1	79.97	91.10	4.4276
Iter2-2	80.18	91.75	4.3570
Iter3-1	79.56	90.63	4.5566
Iter3-2	79.72	90.90	4.5167
Iter3-3	79.90	91.11	4.4798
Iter4-1	79.52	90.49	4.5972
Iter4-2	79.57	90.62	4.5590
Iter4-3	79.67	90.77	4.5468
Iter4-4	79.77	90.93	4.5090

Table 3: Performance comparison of different data utilization strategies. Iter$n$-k denotes the k-th step in the n-iteration setting, where the 12,800 training samples are split into n equal portions.

4.2 Impact of Error Correction Strategy

We try 3 different strategies. "Major-only" means only correcting major errors, "Major&Half Minor" means correcting all major errors and random 50% of minor errors. "Major&Minor" means correcting all erros.

We initially hypothesized that focusing on different types of errors in each iteration might be beneficial, thus exploring various error correction strategies. However, our experimental results suggest otherwise. **In our first-round annotations, we identified 15,879 major errors and 15,555 minor errors for 12,800 translations.** The results demonstrate that partially or completely omitting minor error corrections during preference data construction leads to missed opportunities for learning important translation patterns, resulting in reduced performance. Therefore, we conclude that correcting all errors in each iteration of prefer-

Strategies	Metrics		
	KIWI	XCOMET	METRICX↓
Major-only (Iter 1)	80.20	92.13	4.2573
Major&Minor (Iter 2)	80.39	92.42	4.2444
Major&Half Minor (Iter 1)	80.28	92.20	4.2457
Major&Minor (Iter 2)	80.42	92.44	4.2486
Major&Minor (Iter 1)	80.38	92.40	4.2382
Major&Minor (Iter 2)	80.67	92.66	4.2187

Table 4: Performance comparison of progressive error correction strategies across iterations.

ence data generation is the optimal strategy.

4.3 Benefits of External Preference Data

We examine the impact of preference data composition on DPO effectiveness. For external preference data, we leverage GPT-4o to generate corrections for 12,800 sentences sampled from our augmented MQM training annotations (distinct from the correction data in Model_0 training set). We evaluate three strategies while maintaining a constant total of 12,800 preference pairs: (1) self-supervised preferences only, (2) external preferences only, and (3) an equal mixture of both sources.

Data Composition	Metrics		
	KIWI	XCOMET	METRICX↓
Self-supervised	**0.8038**	0.9240	4.2382
External	0.8032	0.9231	4.2416
Mixed (1:1)	0.8030	**0.9288**†‡	**4.2192**†‡

Table 5: Comparison of preference data strategies: self-supervised preferences, external preferences (human annotations with GPT-4o corrections), and their equal mixture. † indicates $p < 0.05$ compared to Self-supervised; ‡ indicates $p < 0.05$ compared to External.

Results show that the balanced mixture strategy achieves optimal performance with statistically significant improvements in XCOMET and METRICX scores. This success stems from combining two complementary sources: external data provides high-quality reference signals, while self-supervised preferences ensure training stability by aligning with the model's current capabilities. This complementary combination proves more effective than using either source alone.

5 Related Works

5.1 Translation Error Detection and Correction

In span-level error detection, neural models have proven effective in identifying errors within machine translations, as demonstrated by AutoMQM (Fernandes et al., 2023), InstructScore (Xu et al., 2023), and XCOMET (Guerreiro et al., 2023). For error correction, recent advancements involve prompting large language models (LLMs) to suggest new translations, exemplified by TowerAPE (Alves et al., 2024a) and GPT-4 prompting (Raunak et al., 2023). Ki and Carpuat (2024), Xu et al. (2024d) and Treviso et al. (2024) integrate detailed error feedback into post-editing prompts. Specifically, LLMRefine (Xu et al., 2024d) employs "succinct explanations" of fine-grained errors to guide models towards better translations through iterative refinement. xTower (Treviso et al., 2024) utilize error spans annotated by humans or predicted by XCOMET, first explain these errors then give refine translations. Inspired by these works, we construct augmented MQM data for training, enabling our Model^E to provide error explanations and improvement suggestions alongside identifying errors in a reference-free mode. This approach increases the reliability of the identified errors and reduces the difficulty of error correction.

5.2 Preference Data Construction for Machine Translation

Preference data are triplets consisting of user prompts, user-preferred responses, and non-preferred responses. However, there has been limited exploration of how to construct such preference data specifically for machine translation tasks. Xu et al. (2024b) constructed preference data using GPT-4 (OpenAI, 2023) and gold reference for Contrastive Preference Optimization (CPO). Yang et al. (2024b) generate preference datasets using MBR decoding on Multilingual Large Language Models (MLLMs) to favor higher-ranked translations. Agrawal et al. (2024) collect sentence-level quality assessments from professional linguists on LLMs' translations and leverage automatic metrics to recover these preferences. They then use this analysis to curate a dataset. While effective, their approaches face the challenges discussed in Sec 1. To address these issues, our work proposes a method to construct translation preference data at scale using monolingual data, tailored to the model's current capabilities, which effectively enhances the model's translation ability.

6 Conclusion

In this study, we initially explain the shortcomings of previous approaches to constructing translation preference data. Then, we propose SSPO, a self-supervised mechanism that enables LLMs to iteratively generate high-quality translation preference data for DPO training. Our analysis reveals that combining self-generated preference data with external preference data in DPO training leads to superior translation quality. We validate SSPO's effectivenes across multiple language piars, domains and models, demonstrating consistent improvements in translation performance without relying on external human or model annotations. Applying SSPO to 14B parameters large language models (LLMs) achieves comparable or better performance than GPT-4o on FLORES and multi-domain test datasets.

Limitations

We primarily conducted experiments on medium-sized models, while testing on larger or smaller models might reveal different optimization dynamics. Our study also observes relatively few iteration rounds, and a longer-term study could provide deeper insights into the convergence patterns. Additionally, while we employ multiple automatic metrics, the lack of human evaluation means that improvements in metric scores may not perfectly align with actual translation quality as perceived by readers.

Acknowledgments

This paper is supported by the General Program of National Natural Science Foundation of China (62176153). We express our profound appreciation to anonymous reviewers for their helpful suggestions. We also thank Jialong Tang and Haoran Wei for their engaging and valuable discussions that enriched our work.

References

Sweta Agrawal, José G. C. 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. *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. 2024a. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

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. 2024b. Tower: An open multilingual large language model for translation-related tasks. *Preprint*, arXiv:2402.17733.

Anthropic. 2023. Claude 3.5.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André F. T. Martins, Graham Neubig, Ankush Garg, Jonathan H. Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. *Preprint*, arXiv:2308.07286.

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. *Preprint*, arXiv:2104.14478.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Yuxin Fu, Shijing Si, Leyi Mai, Xi-ang Li, and Yulian An. 2024. Ffn: a fine-grained chinese-english financial domain parallel corpus. *arXiv preprint arXiv:2406.18856*.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Preprint*, arXiv:2310.10482.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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. *Preprint*, arXiv:2410.02631.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. *arXiv preprint arXiv:2404.07851*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024. GPT-4o.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Miguel Ramos, Patrick Fernandes, António Farinhas, and Andre Martins. 2024. Aligning neural machine translation models: Human feedback in training and inference. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 258–274, Sheffield, UK. European Association for Machine Translation (EAMT).

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 35053506, New York, NY, USA. Association for Computing Machinery.

Vikas Raunak, Amr Sharaf, Yiren Wang, Hany Awadalla, and Arul Menezes. 2023. Leveraging GPT-4 for automatic translation post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12009–12024, Singapore. Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. 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, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

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).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Marcos Treviso, Nuno M. Guerreiro, Sweta Agrawal, Ricardo Rei, José Pombal, Tania Vaz, Helena Wu, Beatriz Silva, Daan van Stigt, and André F. T. Martins. 2024. xtower: A multilingual llm for explaining and correcting translation errors. *Preprint*, arXiv:2406.19482.

Kaden Uhlig, Joern Wuebker, Raphael Reinauer, and John DeNero. 2025. Cross-lingual human-preference alignment for neural machine translation with direct quality optimization. *Preprint*, arXiv:2409.17673.

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. *Preprint*, arXiv:2309.11674.

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. *Preprint*, arXiv:2401.08417.

Nuo Xu, Jun Zhao, Can Zu, Sixian Li, Lu Chen, Zhihao Zhang, Rui Zheng, Shihan Dou, Wenjuan Qin, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024c. Advancing translation preference modeling with rlhf: A step towards cost-effective solution. *Preprint*, arXiv:2402.11525.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2024d. LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1429–1445, Mexico City, Mexico. Association for Computational Linguistics.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. Instructscore: Explainable text generation evaluation with finegrained feedback. *Preprint*, arXiv:2305.14282.

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, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Guangyu Yang, Jinghong Chen, Weizhe Lin, and Bill Byrne. 2024b. Direct preference optimization for neural machine translation with minimum bayes risk decoding. *Preprint*, arXiv:2311.08380.

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

Vilém Zouhar, Shuoyang Ding, Anna Currey, Tatyana Badeka, Jenyuan Wang, and Brian Thompson. 2024. Fine-tuned machine translation metrics struggle in unseen domains. *arXiv preprint arXiv:2306.07899*.

A Datasets Statistics

A.1 Prompts for Producing Error Annotation and Correction Training Data

Error Annotation. As shown in Figure 4, the template includes 3 examples demonstrating various types of errors and their corrections, followed by the actual task structure. Each error analysis includes location, severity, explanation, and improvement suggestions. For each language pair, we carefully curate examples with diverse error types and varying complexity levels to stimulate the model's error annotation capabilities.

Error Correction. When doing error correction, we use 1-shot prompting in Figure 5. The example should be relatively long with multiple types of errors, serving to stimulate the model's ability to fully correct the translation.

A.2 Translation Data

Language Pair	Quantity	Source
da → en	5000	Sample from Europarl v10
en → da	5000	Sample from Europarl v10
de → en	5000	Sample from News Commentary v18.1
en → de	25227	WMT dev test
fr → en	22074	WMT dev test
en → fr	5000	Sample from news commentary v18.1
id → en	5000	Sample from news commentary v18.1
en → id	5000	Sample from news commentary v18.1
zh → en	16587	WMT dev test
en → zh	11050	WMT dev test

Table 6: Translation Data Statistics

A.3 MQM Data

Table 7 shows the source and distribution of the orginal MQM data.

Language Pair	Quantity	Source
da → en	0	\
en → da	0	\
de → en	3667	(Zouhar et al., 2024)
en → de	7865	WMT Metrics Shared Task
fr → en	3029	(Zouhar et al., 2024)
en → fr	0	\
id → en	0	\
en → id	0	\
zh → en	41943	WMT Metrics Shared Task
en → zh	0	\

Table 7: MQM Data Statistics

Original MQM data contains **severity**, **location** and specific **category** for each error. i.e. each error $\mathbf{e}_i^{\text{original}}$ conatins $\mathbf{loc}_i, \mathbf{sev}_i, \mathbf{cat}_i$.

We augment MQM data by prompting GPT-4o to produce **explanation** and **correct suggestions** for each error. In this process, we intentionally omitted the error type information from the original MQM annotations to simplify the data structure and reduce the complexity of model learning. Each error $\mathbf{e}_i^{\text{augmented}}$ conatins $\mathbf{loc}_i, \mathbf{sev}_i, \mathbf{exp}_i, \mathbf{sugg}_i$.

Fig 6 shows an example of the augmented MQM data.

Our augmentation manner implicitly incorporates a Chain-of-Thought (CoT) mechanism: by requiring the model to first explain the error ($\mathbf{exp}_i$) before generating a correction suggestion ($\mathbf{sugg}_i$), we enforce a step-by-step reasoning process. **The model must understand the error (e.g., semantic mismatch, grammatical flaw, or cultural mistranslation) before proposing a fix**, mirroring the "diagnose-then-correct" workflow of human experts. This idea is also used in the design of xTower(Treviso et al., 2024).

A.4 Error Correction Data

We only construct zh → en and de → en error since it is not a hard task.

Language Pair	Quantity	Source
zh → en	9322	GPT-4o prompting
de → en	9786	GPT-4o prompting

Table 8: Error Detection Data Statistics

A.5 Monolingual Data

We employ a **two-stage dynamic filtering** approach to curate high-quality monolingual data for iterative training:

Length Filtering.

- *Procedure*: When generating preference dataset P_0 from $\mathcal{D}_0$ via Model$_0$, we compute the mean token count $\bar{T}_0$ of source sentences in P_0. This identifies the typical sentence length where Model$_0$ makes errors.
- *Application*: For Model$_1^T$'s training, we sample monolingual sentences longer than $\bar{T}_0$ for translation.
- *Rationale*: Longer sentences offer two advantages: (i) they capture diverse linguistic phenomena (e.g., discourse coherence, idiomaticity), ensuring the model encounters challenging cases; (ii) they mitigate trivial corrections

from short, error-free translations that offer no training signal.

Perplexity (PPL) Filtering.

- *Procedure*: After length filtering, we compute the perplexity of the remaining monolingual corpus using the current iteration's translation model (e.g., Model_1^T). We retain sentences with PPL $\leq \mu + 2\sigma$, where μ denotes the mean PPL and σ the standard deviation.

- *Rationale*: This threshold serves dual purposes: (i) it excludes outliers such as overly complex or noisy sentences beyond the model's current capability; (ii) it balances difficulty by ensuring sentences are challenging yet interpretable, avoiding degenerate cases such as garbled text.

B Experimental Details

Here we list additional experimental details for our implementation and experiments.

B.1 Training Configurations

Here we detail our prompts, training parameters and implementation environment.

B.1.1 Prompts

The followings are our prompts during model training and inference.

Translation Generation Prompt

```
Translate the following {src_lang} text into {tgt_lang}.
{src_lang}: {src}

{tgt_lang}:
```

Error Annotation Prompt

```
Based on the {src_lang} source, identify the major and
minor errors in the {tgt_lang} translation. For each
error, please provide explanation and improvement.
{src_lang} source:{src}
{tgt_lang} translation:{trans}

Errors:
```

Error Correction Prompt

```
Given the {src_lang} source text, the initial {tgt_lang}
translation, and the list of identified errors with
explanations and suggested improvements, improve
the initial {tgt_lang} translation to make it accurate,
fluent, and true to the meaning and tone of the original
text.
{src_lang} source:{src}
Initial {tgt_lang} translation:{initial_trans}
Errors :{errors}

Improved Translation:
```

B.1.2 DPO Training

We use SWIFT framework (Zhao et al., 2024) with the following parameter setting in Table 9. We use 8 80G A100 GPUs for 50 hours DPO training for all our models.

Parameter	Value
Training Type	DPO with LoRA
DPO Config	β=0.1, α=1.0
LoRA Config	rank=128, α=16, dropout=0.1
Learning Rate	1e-5 (cosine schedule)
Sequence Length	1024
Optimization	weight_decay=0.1, max_grad_norm=1.0

Table 9: DPO Training Configuration

B.1.3 SFT Training

We use Deepspeed framework (Rasley et al., 2020) with the following parameter setting in Table 10. We use 8 80G A100 GPUs for 100 hours training for all our models.

Parameter	Value
Training Type	SFT with LoRA
Deepspeed Config	zero_stage=0
LoRA Config	rank=128, α=32, dropout=0.1
Sequence Length	1024
Learning Rate	1e-4(cosine schedule)
Optimization	weight_decay=0.1, max_grad_norm=1.0

Table 10: SFT Training Configuration

B.2 zh $\rightarrow$ en Multi-domain Test set

We collect multi-domain dataset from various sources, as shown in Table 11

Domain	Count	Source
Industry	3,487	(Hu et al., 2024)
Talk	2,599	IWSLT 16,17
IT	2,293	(Hu et al., 2024)
News	1,875	WMT22 New Task Test Set
Literary	1,514	WMT24 Literary Task Test Set
Finance	1,322	(Fu et al., 2024)
E-commerce	1,001	(Hu et al., 2024)
Thesis	625	Sample from (Tian et al., 2014)
Biology	575	WMT Biodemical Translation Task
Science	503	Sample from (Tian et al., 2014)

Table 11: Distribution of test samples across different domains for zh$\rightarrow$en direction

B.3 Multi-lingual Test Set

B.4 Full Multi-domain Results

We show the detailed zh $\rightarrow$ en multi-domain results in Table 13.

FLORES-200 Test Set		
Source → Target	Language Pair	Size
da ↔ en	Danish ↔ English	1,012
de ↔ en	Germen ↔ English	1,012
fr ↔ en	French ↔ English	1,012
id ↔ en	Indonesian ↔ English	1,012
en → zh	English → Chinese	1,012

Table 12: Test set composition from FLORES-200 across different language directions

B.5 Model Evaluation

Evaluation Prompt We use the following the following prompt on Claude-3.5 to evaluate translation quality.

Translation Evaluation Prompt

```
Chinese Source: {src}
Translation 1: {trans1}
Translation 2: {trans2}

Please evaluate the translation quality from the
following aspects:
1. Accuracy: Whether the translation accurately
conveys the meaning of the original text.
2. Fluency: Whether the translation is natural and
idiomatic English.
3. Fidelity: Whether the translation is faithful to the
original without adding or omitting information.
4. Style and Tone: Whether the translation maintains
the style and tone of the original text.

After considering all these factors, please indi-
cate which translation is better:
Reply "1" if Translation 1 is better
Reply "2" if Translation 2 is better
Reply "0" if they are equally good

Please only reply with the number without any
explanation.
```

Evaluation Results As shown in Table 14, Qwen-Model$_1^T$ outperform xTower Pipeline in both evaluation orders, reaching a net win rate of 8.5%.

B.6 Full Multilingual Result

Full results for xx → en are in Table 15. Full results for en → xx are in Table 16.

C Evolvement of Model^E

We focus on zh → en language pair (as our multi-domain test set provides more representative results) and analyze the evolvement of Model^E from three perspectives. We use Qwen series for our analysis.

C.1 Error Rates During SSPO Iterations

We calculated the error rates identified during the two iterations of SSPO training. As shown in Table 18, the error rate decreases from **48.4%** in the first iteration to **37.0%** in the second iteration. **This reduction can be attributed to the improved translation quality of Qwen-Model$_1^T$.**

C.2 MQM Pattern Error Detection

We evaluate Model^E's error deetction performance using the WMT22 zh → en MQM dataset, which consists of 29,579 sentences. We employ Qwen-Model$_0$, Qwen-Model$_1^E$, and Qwen-Model$_2^E$ as the error detection models to annotate errors then calculate precision, recall, and F1 scores. We calculate by the following settings.

TP (True Positive): The model correctly identifies errors or agrees with the test set that the translation is error-free.

FP (False Positive): The model predicts an error where none exists.

FN (False Negative): The model fails to detect an actual error.

Results in Table 18 reveals significant evolvement for the error detection ability of Model^E:

- **Substantial improvement in precision**: From 34.30% in Model$_0$ to 51.90% in Model$_2^E$ (+17.60%), indicating a more accurate understanding of "actual translation error."
- **Steady growth in recall**: From 50.28% in Model$_0$ to 61.09% in Model$_2^E$ (+10.81%), suggesting the model's ability to identify a wider range of translation errors.
- **Overall improvement in F1 score**: From 40.78% in Model$_0$ to 56.12% in Model$_2^E$ (+15.34%), reflecting a better balance between precision and recall.

We find that the training MQM data for Model$_0$ is imbalanced, with error-containing samples significantly outnumbering error-free ones. This leads to over-predictions. **The error annotation datasets curated from our SSPO framework is much more balanced, mitigates the over-prediction issue and lead to more calibrated predictions.**

C.3 Multi-domain Error Detection And Correction

We use Qwen2.5-14B-Instruct as the translation model to generate initial translations on our multi-domain Zh-En test set. We then apply Qwen-Model$_0$, Qwen-Model$_1^E$, and Qwen-Model$_2^E$ to an-

Domain	KIWI			XCOMET			METRICX↓		
	Model_0	Model_1^T	Model_2^T	Model_0	Model_1^T	Model_2^T	Model_0	Model_1^T	Model_2^T
E-commerce	63.90	64.89	65.12	77.68	84.86	85.53	6.6998	6.1222	6.0991
Industry	81.67	82.63	82.79	91.46	93.33	93.47	4.2090	3.6912	3.7428
IT	84.06	84.49	84.75	94.61	95.66	95.92	3.5555	3.2356	3.2253
Literary	72.96	74.66	75.01	86.04	88.23	88.49	5.8166	5.4268	5.3986
Science	86.13	86.60	86.98	96.55	97.06	97.26	3.5208	3.2702	3.2741
Thesis	79.43	80.24	80.37	88.97	90.86	90.95	4.0493	3.5893	3.6219
News	77.18	78.65	78.96	91.06	93.39	93.71	4.6709	4.1914	4.1423
Bio	83.70	84.75	84.93	91.61	93.06	93.28	4.0794	3.7218	3.7051
Talk	82.59	83.82	84.25	96.43	97.09	97.29	5.0868	4.7863	4.6798
Finance	82.56	83.07	83.35	89.32	90.42	90.73	4.7052	4.3470	4.2984

Table 13: Performance comparison across different domains and iterations on zh $\rightarrow$ en (Qwen-Model_0, Qwen-Model_1^T, Qwen-Model_2^T).

Order	Win	Tie	Loss
xTower Pipeline v.s. **Qwen-Model_1^T**	3,870	5,444	6,477
Qwen-Model_1^T v.s. **xTower Pipeline**	4,747	6,371	4,673

Table 14: LLM evaluation results between Qwen-Model_1^T and xTower Pipeline

notate errors and refine the translations. We compare the translation quality and the number of major and minor errors detected by each model.

Results in Table 19 shows that **translation quality improves consistently** across all metrics from Model_0 to Model_2^E. Besides, **error annotation becomes more precise**, as later models identify fewer errors yet achieve better translations. This suggests that Model^E are learning to generalize across domains, identifying more meaningful errors and reducing over-prediction.

D Ablation Study

D.1 Necessity of Model Separation

As depicted in Fig 1, SSPO algorithm separates the initial model Model_0 into two specialized models: Model^E for error detection and Model^T for translation generation. To validate this design, we conducted a controlled experiment by training a hybrid model Qwen-Model_1^{E+T} that combines both capabilities, using the same DPO training data from the first iteration.

Table 20 compares the performance of three models on the zh→en multi-domain test set. The results show a clear performance hierarchy: Qwen-Model_1^T outperforms Qwen-Model_1^{E+T}, which in turn surpasses Qwen-Model_0. While the hybrid model improves upon the baseline, it falls short of the specialized translation model.

This performance gap stems from the conflicting optimization objectives: DPO for translation versus SFT for error correction. Training a single model for both tasks risks catastrophic forgetting, where improving one capability degrades the other. Therefore, separating Model_0 into task-specific models (Model^E and Model^T) proves essential for optimal performance.

D.2 Effectiveness of DPO Training

To validate the effectiveness of our DPO process, we use the positive example in our preference data to SFT Qwen-Trans (the model trained solely on the translation data in A.2) and get Qwen-Trans-SFT_1. We compare its performance with our Qwen-Model_1^T. This experiment is carried out on zh $\rightarrow$ en direction with the preference data used in the first iteratioin.

Results in Table 21 demonstrate that our method outperforms SFT method.

E Potential Risks

This paper presents work whose goal is to advance the field of Machine Translation and Large Language Model. We used open-source data and models to do machine translation tasks without other intend. We don't include offensive information in our data. We think we don't have risks include potential malicious or unintended harmful effects and uses.

Models	da→en			de→en			fr→en		
	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓
TowerInstruct	90.80	94.05	4.8012	89.31	90.50	5.0228	88.86	89.14	5.0336
GPT-4o-0806	**93.49**	**96.96**	**4.4658**	91.46	96.77	2.2697	92.27	95.69	4.7092
Qwen-Model0	92.46	95.36	4.6772	91.19	96.33	2.3186	92.23	95.26	4.8869
Qwen-Model$_1^T$	92.70	95.73	4.6444	91.41	96.52	2.2997	92.66	95.71	4.7578
Qwen-Model$_2^T$	93.38	96.81	4.5120	91.63	96.78	2.2801	**92.83**	**95.89**	**4.6820**
Mistral-Model0	91.15	94.14	4.7889	91.36	96.67	2.3068	92.12	94.81	4.9447
Mistral-Model$_1^T$	92.87	96.44	4.5971	91.58	96.82	2.2710	92.31	95.09	4.7447
Mistral-Model$_2^T$	93.05	96.69	4.5230	**91.73**	**96.98**	**2.2523**	92.54	95.43	4.6980

Models	id→en			zh→en			**Avg.**		
	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓
TowerInstruct	85.06	83.06	5.8541	78.92	90.22	4.7780	86.59	89.39	5.0979
GPT-4o-0806	89.39	96.27	4.7982	80.29	91.54	4.3310	89.38	95.45	4.1148
Qwen-Model0	89.48	96.16	4.8467	79.42	90.37	4.6393	88.96	94.70	4.2737
Qwen-Model$_1^T$	89.49	96.43	4.7885	80.38	92.40	4.2382	89.33	95.36	4.1457
Qwen-Model$_2^T$	**89.65**	**96.58**	**4.7120**	**80.67**	**92.66**	**4.2187**	**89.63**	**95.74**	**4.0810**
Mistral-Model0	89.23	96.02	4.8796	79.16	90.31	4.7038	88.6	94.39	4.3248
Mistral-Model$_1^T$	89.28	96.13	4.8100	79.91	92.77	4.4024	89.19	95.45	4.1650
Mistral-Model$_2^T$	89.41	96.32	4.7290	80.29	93.08	4.3354	89.40	95.7	4.1075

Table 15: Performance comparison on xx→en translation across different language pairs.

Models	en→da			en→de			en→fr		
	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓
TowerInstruct	86.69	94.81	1.3383	86.19	98.17	0.8424	90.79	96.43	1.1191
GPT-4o-0806	**92.11**	**97.53**	**0.9415**	87.06	98.33	0.8053	**90.88**	**96.57**	**1.1087**
Qwen-Model0	75.37	92.73	1.4860	85.04	97.94	0.8839	89.35	95.53	1.2164
Qwen-Model$_1^T$	83.24	92.99	1.5154	85.13	98.04	0.8772	89.87	95.73	1.2060
Qwen-Model$_2^T$	84.62	93.53	1.3920	85.32	98.18	0.8630	90.36	96.01	1.1680
Mistral-Model0	88.44	96.08	1.1123	86.61	98.41	0.8076	90.23	96.08	1.1450
Mistral-Model$_1^T$	89.13	96.21	1.0786	86.84	98.39	0.7931	90.16	96.21	1.1407
Mistral-Model$_2^T$	89.65	96.49	1.0230	**87.21**	**98.54**	**0.7800**	90.58	96.44	1.1190

Models	en→id			en→zh			**Avg.**		
	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓	KIWI	XCOMET	METRICX↓
TowerInstruct	84.48	92.23	1.7846	86.19	90.66	1.5572	86.87	94.46	1.3283
GPT-4o-0806	**90.43**	**95.97**	**1.2112**	**86.90**	91.57	1.4737	**89.48**	**95.99**	**1.1081**
Qwen-Model0	87.45	94.12	1.4011	85.60	91.19	1.5656	84.56	94.3	1.3106
Qwen-Model$_1^T$	88.26	94.49	1.4094	85.89	91.49	1.4094	86.48	94.55	1.2835
Qwen-Model$_2^T$	88.73	94.82	1.3680	86.35	**91.72**	**1.3720**	87.08	94.85	1.2326
Mistral-Model0	87.45	93.86	1.4874	85.62	90.40	1.5505	87.67	94.97	1.2206
Mistral-Model$_1^T$	87.35	93.95	1.4808	85.88	90.64	1.5212	87.87	95.08	1.2029
Mistral-Model$_2^T$	87.79	94.23	1.4350	86.21	90.98	1.4830	88.29	95.34	1.1680

Table 16: Performance comparison on en→xx translation across different language pairs.

Iteration	#Monolingual	#Erroneous	#Error-free	Error Rate
1 (Qwen-Model$_0$ annotate Qwen-Model$_0$)	39880	19290	20590	48.4%
2 (Qwen-Model$_1^E$ annotate Qwen-Model$_1^T$)	25533	9465	16088	37.0%

Table 17: Error rates during actual SSPO iterations. **#Monolingual** represents the total number of monolingual sentences processed in each iteration. **#Erroneous** shows the count of sentences identified as containing translation errors. **#Error-free** indicates the number of sentences identified to have acceptable translations without errors.

```
# Example1
## Question
source: 'It is particularly satisfying that Ruby has produced a female calf - these are vital for the
 development of the breeding programme and Stella will eventually move to another collection to
hopefully become a successful mother herself.'
translation: Es ist besonders befriedigend, dass Ruby ein weibliches Kalb hervorgebracht hat - diese
sind für die Entwicklung des Zuchtprogramms von entscheidender Bedeutung - und Stella wird schlielich
 in eine andere Kollektion wechseln, um hoffentlich selbst eine erfolgreiche Mutter zu werden".
errors: There is a Major error at "Kollektion". There is a Minor error at "Bedeutung -". There is a
Minor error at "-". There is a Minor error at "".
This is an English-to-German translation error annotation task. Only the location and severity of
errors are provided. Please analyze each error by considering the context in the translation field
and the English source text, and provide explanations and improvement suggestions that conform to
German grammar, conventions and idiomatic usage.

## Answer
[{
    "location": "Kollektion",
    "severity": "major",
    "explanation": "The term 'Kollektion' is not appropriate for referring to a group or place where
animals are kept or bred. In the context of animal breeding programs, 'Sammlung' or 'Zuchtbestand'
would be more suitable.",
    "improvement": "Change 'Kollektion' to 'Sammlung' or 'Zuchtbestand'."
},
{
    "location": "Bedeutung -",
    "severity": "minor",
    "explanation": "The use of a dash here is not typical in German writing and may disrupt the
sentence structure. A comma or simply ending the sentence would be more appropriate.",
    "improvement": "Replace the dash with a comma or end the sentence after 'Bedeutung'."
}]

# Example2
...

# Example3
...

# Your task
## Question
source: {source}
translation: {translation}
errors: {errors}
This is an English-to-German translation error annotation task. Only the location and severity of
errors are provided. Please analyze each error by considering the context in the translation field
and the English source text, and provide explanations and improvement suggestions that conform to
German grammar, conventions and idiomatic usage.

## Answer
```

Figure 4: Prompt template for en $\rightarrow$ de translation error annotation. The template demonstrates various error types including terminology misuse, punctuation errors, and grammatical issues, with detailed explanations and improvements for each error.

Model	Precision	Recall	F1	TP	FP	FN
Qwen-Model$_0$	0.3430	0.5028	0.4078	18473	35381	18267
Qwen-Model$_1^E$	0.5119	0.5935	0.5497	22638	21588	15505
Qwen-Model$_2^E$	**0.5190**	**0.6109**	**0.5612**	**22954**	**21270**	**14623**

Table 18: Error detection results on WMT22 zh $\rightarrow$ en MQM test set.

```
## Question
Given the English source text, the initial German translation, and the list of identified errors with
 explanations and suggested improvements, improve the initial German translation to make it accurate,
 fluent, and true to the meaning and tone of the original text.

English source: Google is marking its own birthday today with a doodle. The doodle features a photo
of a 90s' style computer with Google search page and date stamp for September 27, 1998, surrounded by
 confetti in Google colours. A smaller doodle with number 21 as candles, as a part of Google logo,
shows on the search results page. The search giant was founded 21 years ago in 1998 by Larry Page and
 Sergey Brin, then students at Stanford University in California. September 27 is, however, not
Google's actual birthday.

Initial German translation: Google feiert heute seinen eigenen Geburtstag mit einem Doodle. Das
Doodle zeigt ein Foto eines Computers im Stil der 90er mit Google-Suchseite und Datumsstempel für den
 27. September 1998, umgeben von Konfetti in Google-Farben. Ein kleineres Doodle mit der Nummer 21
als Kerzen als Teil des Google-Logos zeigt sich auf der Suchergebnisseite. Der Suchriese wurde 1998
vor 21 Jahren von Larry Page und Sergey Brin, damals Studenten an der Stanford University in
Kalifornien, gegründet.

Errors: [{
    "location": "90er",
    "severity": "minor",
    "explanation": "While '90er' is commonly used, it should be followed by 'Jahre' to fully convey
the meaning of '90s' style.",
    "improvement": "Change '90er' to '90er Jahre' for clarity."
},
{
    "location": "gegründet.",
    "severity": "minor",
    "explanation": "The final sentence about September 27 not being Google's actual birthday is
missing.",
    "improvement": "Add 'Der 27. September ist jedoch nicht Googles eigentlicher Geburtstag.' at the
end."
}]

Improved Translation: Google feiert heute seinen eigenen Geburtstag mit einem Doodle. Das Doodle
zeigt ein Foto eines Computers im Stil der 90er Jahre mit Google-Suchseite und Datumsstempel für den
27. September 1998, umgeben von Konfetti in Google-Farben. Ein kleineres Doodle mit der Nummer 21 als
 Kerzen als Teil des Google-Logos zeigt sich auf der Suchergebnisseite. Der Suchriese wurde 1998 vor
21 Jahren von Larry Page und Sergey Brin, damals Studenten an der Stanford University in Kalifornien,
 gegründet. Der 27. September ist jedoch nicht Googles eigentlicher Geburtstag.

# Your task
Given the English source text, the initial German translation, and the list of identified errors with
 explanations and suggested improvements, improve the initial German translation to make it accurate,
 fluent, and true to the meaning and tone of the original text.

English source: {source}
Initial German translation: {translation}
Errors: {errors}
Improved Translation:
```

Figure 5: Prompt template for translation improvement task. The template shows how to analyze translation errors and make improvements while maintaining accuracy and fluency in German. The example demonstrates handling of missing content and stylistic refinements.

Model	KIWI	XCOMET	METRICX↓	#Major	#Minor
Qwen2.5-14B-Instruct	78.52	90.54	4.8249	\	\
Qwen-Model$_0$ Refine	78.53	91.28	4.7158	7906	11927
Qwen-Model$_1^E$ Refine	79.00	91.64	4.6537	4340	3856
Qwen-Model$_2^E$ Refine	**79.21**	**91.82**	**4.6329**	4832	3319

Table 19: Error annotation and improvement results of different Model^E on the initial translations generated by qwen2.5-14b-instruct for the zh→en multi-domain test set.

```
{
"source":" 市内一座商场同样倒塌，数百名居民赶到现场，等候亲友的音讯。",
"translation":"A shopping mall collapsed, and hundreds of
residents rushed to the scene, waiting for the audio of friends
and relatives.",
"original errors":"There is a major error at \"audio\"",
"augmented errors":[{
"location": "audio",
"severity": "major",
"explanation": "The term '音讯' in the original text is
misinterpreted as 'audio'. In this context, '音讯' means 'news'
or 'information' regarding the safety of friends and relatives,
not 'audio'.",
"improvement": "Change 'audio of friends and relatives' to
'news of friends and relatives'."
}]
}
```

Figure 6: An example of the augmented MQM data. Compared to the orginal MQM data, wo add explanations for each error and provide improvement suggestions.

Model	KIWI	XCOMET	METRICX↓
Qwen-Model$_0$	79.42	90.37	4.6393
Qwen-Model$_1^{E+T}$	80.01	91.50	4.4495
Qwen-Model$_1^T$	**80.38**	**92.40**	**4.2382**

Table 20: Performance comparison of model separation strategy.

Model	KIWI	XCOMET	METRICX↓
Qwen-Trans-SFT$_1$	80.22	91.56	4.3690
Qwen-Model$_1^T$	**80.38**	**92.40**	**4.2382**

Table 21: Performance comparison between Qwen-Trans-SFT$_1$ and Qwen-Model$_1^T$. The best results are in **bold**.