Plan2Align: Predictive Planning Based Test-Time Preference Alignment in Paragraph-Level Machine Translation

Kuang-Da Wang^{1,†} Teng-Ruei Chen^{1,†} Yu Heng Hung¹ Shuoyang Ding² Yueh-Hua Wu² Yu-Chiang Frank Wang² Chao-Han Huck Yang² Wen-Chih Peng¹ Ping-Chun Hsieh¹

¹National Yang Ming Chiao Tung University ²NVIDIA

 $\label{lem:cs10@} $$\{gdwang.cs10@,raychen.cs12@,wcpeng@cs.,pinghsieh@\}nycu.edu.tw $$\{hucky,krisw,shuoyangd\}@nvidia.com $† equal contribution $$$$

Abstract

Machine Translation (MT) has been predominantly designed for sentence-level translation using transformer-based architectures. While next-token prediction based Large Language Models (LLMs) demonstrate strong capabilities in long-text translation, non-extensive language models often suffer from omissions and semantic inconsistencies when processing paragraphs. Existing preference alignment methods improve sentence-level translation but fail to ensure coherence over extended contexts due to the myopic nature of next-token generation. We introduce Plan2Align, a test-time alignment framework that treats translation as a predictive planning problem, adapting Model Predictive Control to iteratively refine translation outputs. Experiments on WMT24 Discourse-Level Literary Translation show that Plan2Align significantly improves paragraph-level translation, achieving performance surpassing or on par with the existing training-time and test-time alignment methods on LLaMA-3.1 8B.

1 Introduction

Machine Translation (MT) has primarily leveraged transformer-based encoder-decoder neural architectures, as exemplified by models like M2M-100 (Fan et al., 2021) and MT5 (Xue et al., 2021). However, the emergence of Large Language Models (LLMs), such as the GPT series (Brown et al., 2020; Achiam et al., 2023), LLaMAs (Touvron et al., 2023a,b), and Gemma (Team et al., 2024), has demonstrated remarkable efficacy in a wide range of NLP tasks (Stiennon et al., 2020; Hendrycks et al., 2021; Srivastava et al., 2023; Yu et al., 2024; Zhong et al., 2024), including multilingual translation. This progress has led to increasing interest in leveraging LLMs for MT, particularly for handling long-form text. LLMs excel at long-text translation due to their ability to model dependencies across extended contexts, enabling improved coherence and fluency. Models such as Tower (Alves et al., 2024) and

GPT-4o (OpenAI, 2024) have demonstrated stateof-the-art performance, surpassing traditional MT systems in both fluency and contextual consistency.

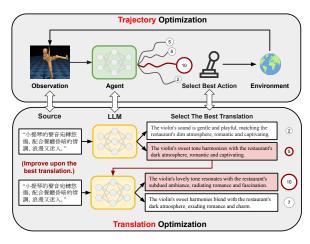


Figure 1: Machine translation via predictive planning from the perspective of trajectory optimization.

However, non-extensive (i.e., smaller than 10B) LLMs often struggle with long-text translation, exhibiting sentence omissions, semantic inconsistencies, and hallucinations (Wu et al., 2024). These limitations stem from the myopic nature of nexttoken prediction, where local fluency is prioritized at the cost of global coherence, leading to degraded translation quality in extended discourse. Onepass translation remains inherently limited, as it lacks iterative refinement, making it prone to overtranslation, under-translation, and compounding errors (Mandya et al., 2020). While traditional MT systems incorporate techniques such as reranking, iterative decoding, and constrained beam search to improve coherence and fidelity, LLMbased MT operates purely in a next-token prediction paradigm, lacking an explicit self-correction mechanism (Hadj Ameur et al., 2019; Feng et al., 2024). As a result, errors introduced early in the generation process propagate across extended text spans, leading to semantic drift, loss of contextual

dependencies, and degraded translation quality in long-form content.

Recent research has investigated preference alignment as a means to improve translation quality in small-scale LLMs, leveraging preference data to guide optimization. Methods such as CPO (Xu et al., 2024b) and WAP (Wu et al., 2024) have shown promising results in aligning translations to preferences. However, existing methods face two fundamental challenges in long-text translation:

- 1. Preference alignment techniques operate at specific granularity levels; when applied at the long-text level, they often struggle to preserve sentence coherence across paragraphs.
- 2. Unlike training-time alignment, which requires costly fine-tuning, testing-time alignment methods must incorporate a technique to recover from past errors or low-quality answers caused by compounding errors during inference. This is crucial because these past mistakes can lead to further errors, as earlier responses influence the generation of answers to subsequent questions (Mandya et al., 2020).

To address these challenges, we propose **Plan2Align**, a test-time alignment method that employs a self-rewriting framework based on the principle of Model Predictive Control (MPC), a classic predictive planning method for trajectory optimization (Rao, 2014) widely used in robotics (Tassa et al., 2012, 2014; Nagabandi et al., 2020) and motion planning (Ji et al., 2016). Rather than fine-tuning LLMs, Plan2Align iteratively refines translations by integrating high-quality segments from previous iterations at test time, enhancing coherence and fluency in long-text translation.

This study focuses on **paragraph-level MT**, where translation quality is evaluated beyond isolated sentences. Specifically, we define *context-level* as a group of adjacent, semantically related sentences, while a paragraph consists of multiple interconnected contexts. Ensuring high-quality translation at the context level is crucial for maintaining coherence and accuracy at the paragraph level. Specifically, this study is twofold: (i) to develop a test-time alignment method that matches or surpasses fine-tuning approaches, and (ii) to enhance paragraph-level translation by improving consistency and fidelity across context-level translations.

As a predictive planning method, MPC optimizes a sequence of decisions by continuously

adjusting future actions based on past observations (Camacho and Bordons, 2007; Kouvaritakis and Cannon, 2016). Inspired by this, we draw an analogy between *trajectory optimization* and *translation optimization*, where prior translations inform subsequent refinements, and recast MT as an iterative optimization process rather than a one-pass mapping. Figure 1 illustrates this correspondence, highlighting the intuition behind our approach. While LLMs excel in contextual understanding, their one-pass translations remain suboptimal. By iteratively leveraging high-quality past translations, our method enhances coherence and discourse integrity, improving overall quality.

However, directly applying MPC to paragraphlevel MT is impractical when entire paragraphlevel translation histories are fed into small-scale LLMs due to their limited contextual capacity. To adapt MPC to paragraph-level MT, we introduce two novel mechanisms: (i) Model-predictive context alignment: We modify MPC to selectively retain high-quality contexts from multiple paragraphlevel translations with the help of a context buffer, instead of relying on a single best trajectory. This ensures effective accumulation and utilization of translation experience for aligning source and translation contexts. (ii) Self-rewriting tasks: We redefine MT as a self-rewriting task, a structured prompting mechanism that enhances discourse coherence, enables each iteration to build upon prior high-quality translations, and ultimately improves fluency and consistency. By integrating these two strategies, Plan2Align achieves efficient translation refinement within a limited number of iterations.

To validate our approach, we apply Plan2Align to LLaMA-3.1 8B on the WMT'24 Discourse-Level Literary Translation Benchmark, using 6K paragraph-level preference data for test-time alignment. Our method demonstrates substantial improvements, achieving performance on par with or surpassing training-time alignment methods while outperforming existing test-time alignment approaches. Plan2Align bridges the gap in LLMbased paragraph-level translation, introducing a novel test-time alignment framework that ensures high translation quality while maintaining contextlevel consistency. Moreover, our test-time alignment method can augment other fine-tuning approaches, leveraging a small number of iterations at test time to achieve even better translation performance.

2 Related Work

2.1 Document-Level Machine Translation

Document-level neural machine translation can be broadly divided into two categories: the sentenceto-sentence (sen2sen) approach and the documentto-document (doc2doc) approach (Maruf et al., 2022; Wang et al., 2023). Translation on a sentenceby-sentence basis, even with augmented contexts, remains inadequate for fully capturing coherence and consistency within texts (Fernandes et al., 2023). Various methods demonstrate the importance of context in translation; (Wu et al., 2023) process context within different components of the same encoder, while (Voita et al., 2018) use attention distributions to integrate context awareness. We are exploring a translation method that effectively bridges the gap between sentence-level and document-level approaches, aiming to harness more contextual information than sentence-level translation while avoiding the extensive resource demands of full document-level translation.

2.2 Preference Alignment for LLMs

Large language models (LLMs) often misalign with human preferences. Traditional approaches such as supervised fine-tuning (SFT) (Ziegler et al., 2019), reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), direct policy optimization (DPO) (Rafailov et al., 2023), and simple preference optimization (SimPO) (Meng et al., 2024) refine models post-training but demand significant computational resources. Testtime alignment offers a lightweight alternative by modifying outputs during inference without altering model parameters. Prior methods include InferAligner (Wang et al., 2024b) for cross-model guidance, ARGS (Khanov et al., 2024) with dynamic reward-based adjustment, TreeBoN (Qiu et al., 2024) leveraging tree-search in Best-of-N Sampling, and RAIN (Li et al., 2024) and URIAL (Lin et al., 2024) employing advanced prompt engineering. However, these approaches struggle when early responses are suboptimal, hindering subsequent outputs. To address this, Plan2Align introduces an MPC framework that enables predictive planning, effectively accumulating and leveraging high-quality experience to optimize response generation. By integrating structured foresight into test-time alignment, Plan2Align mitigates earlystage errors, leading to more reliable and adaptable model outputs.

3 Methodology

In this section, we introduce the Plan2Align framework (Figure 2). We describe how predictive planning enables preference alignment, the operational sequence of the framework, and how final translations are produced.

3.1 Test-Time Preference Alignment via Predictive Planning

Our key idea is to recast MT as trajectory optimization and thereby employ predictive planning for iterative translation refinement.

Machine Translation as Trajectory Optimiza-

tion. The goal of trajectory optimization is to find an optimal sequence of actions $\mathbf{a}^* = (a_1^*, \cdots, a_T^*)$ such that the expected trajectory-wise utility is maximized (Chua et al., 2018; Lowrey et al., 2019). More specifically: Let s_t denote the state of the environment at the t-th step and let s_1 be the initial state. Consider an environment with Markov transition dynamics, where the state s_{t+1} is independent of the past history given the current state s_t and action a_t . Let $\mathcal{J}(a,s)$ denote the utility function of the sequences a and s. Then, the search for a^* can be formulated as

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}_{1:T}} \mathbb{E}[\mathcal{J}(\boldsymbol{a}_{1:T}, \boldsymbol{s}_{1:T})|s_1],$$
 (1)

where T is the planning horizon and the expectation is over all the randomness in the environment.

One can interpret translation as an instance of trajectory optimization by (i) treating the source paragraph as the initial state s_1 , (ii) treating each a_t as the t-th segment of the translation, and (iii) setting s_{t+1} as the concatenation of the source paragraph and the translations up to the t-th segment.

Model Predictive Control. In general, direct optimization of (1) requires searching over all possible action sequences of length T and is computationally intractable. As a predictive planning method, MPC approximately solves (1) by iteratively solving local optimization problems (Hansen et al., 2022), instead of globally optimizing the total utility in one pass. Specifically, MPC sequentially determines the action of each time step t by estimating the optimal subsequence $a_{t:t+H}^*$ given the current state s_t , i.e.,

$$\Pi^{\text{MPC}}(s_t) = \arg \max_{\boldsymbol{a}_{t:t+H}} \mathbb{E} [\mathcal{J}(\boldsymbol{a}_{t:t+H}, \boldsymbol{s}_{t:t+H}) | s_t],$$
(2)

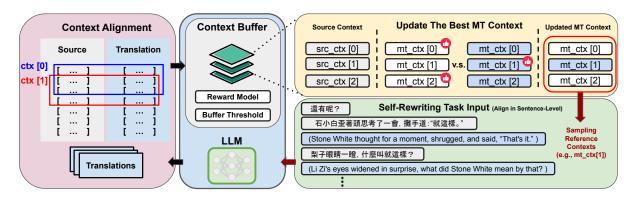


Figure 2: The framework of Plan2Align. To leverage past high-quality translations for preference alignment, Plan2Align transforms the translation task into a self-rewriting process. The LLM refines its translations by revisiting previous outputs, segmenting the paragraph-level result, and aligning at the context level. A reward model, trained on preference data, selects optimal contexts to update the context buffer. Through iterative refinement, the better context buffer improves context quality and generates better rewriting inputs, achieving both paragraph-level preference alignment and context-level coherence.

and then executing the first action of $\Pi^{\text{MPC}}(s_t)$. In practice, MPC solves (2) by employing a learned predictive dynamics model to generate multiple H-step predictive rollouts $\{(\boldsymbol{a}_{t:t+H}^{(i)}, \boldsymbol{s}_{t:t+H}^{(i)})\}_{i=1}^K$ and obtain an approximate maximizer by taking the maximum of $\mathcal{J}(\boldsymbol{a}_{t:t+H}^{(i)}, \boldsymbol{s}_{t:t+H}^{(i)})$ over the rollouts.

Predictive Planning for Preference Alignment.

In paragraph-level MT, we adopt the principle of predictive planning in MPC for preference alignment. Figure 2 illustrates the framework of Plan2Align. Specifically: (i) We train a reward model on preference data. This reward model, which plays the role of utility function in MPC, assesses translation context quality and serves as a guiding function to align paragraph-level translations with preferences. By integrating MPC, we iteratively refine translations, ensuring alignment with both linguistic quality and contextual consistency. (ii) The LLM itself can serve as a predictive dynamics model as the next state can be obtained by appending the newly generated translation segment to the current state. (iii) To enable MPC for paragraph-level MT while ensuring strong contextlevel performance, we further introduce two key mechanisms for accumulating and utilizing highquality translation experiences as described in the next two subsections.

3.1.1 Context Alignment and Context Buffer

A fundamental challenge in using MPC in paragraph-level MT is: *How to continuously accumulate and retain valuable translation experience?* A high-scoring translated paragraph does not necessarily indicate that every part of it is of high quality.

Moreover, non-extensive LLMs struggle to identify and leverage high-quality contexts from translated paragraphs due to their limited long-text comprehension capabilities.

Instead of direct MPC application in MT, we modify MPC to selectively retain high-quality contexts from multiple paragraph-level translations instead of relying on a single best trajectory. This ensures effective accumulation and utilization of translation experience. To achieve this, we establish a **context buffer**, which stores high-quality context extracted from multiple paragraph-level translation experiences.

In our implementation, each context unit consists of three sentences, represented in the buffer as key-value pairs in a dictionary structure: source contexts (src_ctx) as keys and corresponding translation contexts (mt_ctx) as values. To comprehensively capture contextual dependencies, the context buffer is constructed using a sliding-window mechanism. Specifically, for a given paragraph, the first three sentences form src_ctx[0], while sentences 2-4 constitute src_ctx[1], and so forth. The translation contexts are structured in the same manner.

Each translation is aligned through *context align-ment*, establishing correspondences between source and translation contexts. We evaluate the aligned source and translation context pairs using a reward model trained on given preference data. Our goal is to ensure that the context buffer retains only high-quality contexts. The reward model assesses candidate context-level translations and selectively retains only the best ones. When a selected translation context surpasses a predefined *buffer threshold*

and a matching source context already exists in the context buffer, a comparison is performed. If the new context falls below the threshold or does not surpass the quality of the existing context in the buffer, it is not stored.

Through this approach, in each iteration, the system filters high-quality contexts, continuously refining the context buffer and ensuring that subsequent translations improve over time.

3.1.2 Translations from Self-Rewriting Tasks

To leverage high-quality contexts in the context buffer for translation improvement, we frame the translation task as a *self-rewriting task*. To increase the diversity of translation outputs generated by the LLM, thereby enriching the context buffer with more varied contexts, we sample *reference contexts* from the buffer to construct the self-rewriting task input, rather than using the entire buffer content. By providing LLMs with pre-aligned content as input for self-rewriting, they can refine paragraphlevel translations using past high-quality contexts while maintaining context-level coherence.

3.2 Final Translation from Context Buffer

The final translation generation differs from context buffer updates. In this phase, the stored high-quality contexts, covering nearly the entire paragraph, serve as reference translations. Unlike the self-rewriting task, which aligns individual sentences, this final translation generation stage constructs the reference translation by incorporating all relevant content from the context buffer and feeding it along with the source paragraph into the LLM. The LLM then improves the reference translation based on the source, preventing overlapping translation artifacts and ensuring fluency. Since this step involves refining a single high-quality reference translation with the source text, even non-extensive LLMs can handle the task effectively.

3.3 The Details of Context Alignment

Since the final translation relies on the quality of the context buffer, improving its quality is crucial. Therefore, continuously updating the context buffer to retain the most valuable translation experience is essential. To achieve this, context alignment must ensure proper pairing between source and translation contexts, allowing the reward model to effectively filter out low-quality translations and retain only the most reliable ones.

However, there is no well-established method to effectively extract contextual information in paragraph-level MT. To address this, we propose a simple yet effective approach for context alignment, which consists of three key steps: (i) Sentence Segmentation: We utilize the SpaCy (Honnibal et al., 2020) toolkit to segment paragraphs into a list of sentences. (ii) Sentence Alignment: We use VecAlign (Thompson and Koehn, 2020) to dynamically search for the optimal alignment between sentences. Since LLMs may introduce hallucinations, leading to sentence omissions or over-translations (resulting in extra sentences), we insert any unaligned translation sentences into their original positions and add empty strings on the source side as placeholders. (iii) Contextual Sliding Window Construction: To capture the contextual structure at the paragraph level, we adopt a sliding window approach with a window size of 3, forming the context-level aligned source-translation pairs.

4 Experiments

4.1 Translation Dataset

Compared to sentence-level MT, paragraph-level MT has received significantly less attention, and many translation models struggle to process longcontext inputs. For example, Post and Junczys-Dowmunt (2024) investigated a long-context training paradigm for machine translation but restrict the maximum sequence length to 256. ALMA (Xu et al., 2024a,b) restricted their post-training datasets to a maximum sequence length of 512. To address this limitation and ensure a suitable benchmark for paragraph-level MT, we use the latest WMT'24 Discourse-Level Literary Translation benchmark (Wang et al., 2024a) for our experiments. This dataset allows for evaluating how well translation methods preserve contextual relationships during translation. The available language pairs include: Chinese →English, Chinese \rightarrow German, and Chinese \rightarrow Russian.

To ensure that the input length remains within the context window of our selected LLMs, we segment each instance from the training and validation sets into chunks of up to 1024 tokens. This preprocessing ensures that paragraph-level MT tasks are effectively handled within the model's constraints. But as the readers shall see, our methodology is easy to extend to even longer context given LLM translation capability is up to par.

		$zh \to en$			$zh \to ru$			$zh \to de$					
Methods	Test-Time	CW-COMET↑	CW-KIWI↑	d-BLEU↑	1-0/0-1 Ratio ↓	CW-COMET ↑	CW-KIWI ↑	d-BLEU↑	1-0/0-1 Ratio ↓	CW-COMET↑	CW-KIWI↑	d-BLEU ↑	1-0/0-1 Ratio ↓
GPT-40 2024-08-06		94.58	73.06	27.77	0.10	93.74	54.20	19.89	0.00	94.54	54.55	21.93	0.00
Qwen-2.5 (14B)	-	94.43	72.44	25.14	0.18	90.47	50.13	18.09	3.08	92.98	50.02	19.30	1.24
Llama-3.1 (8B)	×	84.36	58.27	20.74	10.47	86.28	34.32	8.30	4.19	88.97	39.49	11.05	4.43
Llama-3.1 _{SFT}	×	93.54	67.16	22.28	0.34	89.11	43.99	13.59	1.92	93.47	47.39	16.12	0.19
Llama-3.1 _{SimPO}	×	91.74	62.31	20.20	1.66	84.56	41.66	11.30	2.53	93.40	46.47	14.39	0.00
Llama-3.1 _{DPO}	×	90.23	62.09	20.50	1.33	82.15	38.91	11.69	6.62	93.48	46.09	14.20	0.00
Llama-3.1 _{Plan2Align_{SFT}}	-	93.05	68.27	21.53	1.46	<u>91.71</u>	41.09	10.52	2.16	<u>93.94</u>	43.41	12.12	0.00
Llama-3.1 _{RAIN}	✓	58.52	36.39	13.60	37.18	66.29	31.75	7.93	27.79	67.43	31.69	10.38	27.15
Llama-3.1 _{ARGS}	✓	63.99	41.62	11.09	31.53	43.03	19.57	2.78	32.96	51.97	23.23	3.76	40.01
Llama-3.1 _{Best-of-60}	✓	90.97	65.41	21.66	3.58	84.86	48.93	5.02	3.89	82.74	38.55	10.78	10.78
Llama-3.1 _{Vanlia MPC}	✓	89.79	58.40	7.30	4.55	82.83	35.18	7.19	10.26	83.67	38.46	10.82	10.63
Llama-3.1 _{Plan2Align}	✓	94.13	67.06	21.17	0.20	91.23	40.11	10.69	1.49	92.60	44.62	14.23	1.44

Table 1: Results on the test set of WMT24 literary translation shared task (zh \rightarrow xx translation directions). Results are separated into three groups: SoTA and base models, training-time alignment methods, test-time alignment methods. For the two groups showing results from alignment methods, best numbers in respective groups are **boldfaced**, and best numbers among all alignment methods are also **underlined**. Proposed methods are highlighted.

4.2 Preference Data and Model Training

Preference data is crucial for training alignment methods and for training the reward model used in Plan2Align. The preference data is derived from the training set of the dataset. Specifically, each instance is segmented into paragraphs of up to 1024 tokens. From each translation direction, we sample 2K paragraphs, resulting in a total of 6K paragraphs for constructing the preference data. We generate translation outputs using LLaMA-3.1-8B-Instruct, Gemma-2-9B, and GPT-4o. The translations are then evaluated using MetricX-24-XL (Juraska et al., 2024), a reference-free quality estimation (QE) model. The QE scores help us select high-quality translations: the translation with the highest score is designated as the preferred translation, while the one with the lowest score is considered not preferred. The translation with a middle score is disregarded. The resulting preference data is used to train the reward model in Plan2Align, enabling it to assess context quality. The trained reward model achieves a final accuracy of 88.53% on the validation set. Further details on the formation of preference data can be found in Appendix A.

4.3 Evaluation Metrics

We focus on optimizing LLM-based paragraph-level MT models to mitigate contextual inconsistencies, hallucinations, and omissions. However, while model-based metrics have taken over string-based metrics as the state-of-the-art, all model-based metrics (e.g. COMET) are still only trained with sentence-level data, and machine translation evaluation on paragraph-level remains an open question. A prior work (Deutsch et al., 2023) argued that existing metrics generalize well to paragraph-level, but the test set in that work is

a mere concatenation of existing datasets translated by sentence-level models, and does not take into account the context-level improvements we are aiming at. In fact, in our preliminary study, we observed serious pathological behaviors of these metrics when evaluating our paragraph-level outputs (see Appendix E). This calls for a significant rework of the existing evaluation paradigm.

The changes we've made to our machine translation evaluation paradigm are as follows:

Contextual Sliding Window. Instead of simply feeding the source, translation and reference sentences into COMET¹ and COMET-KIWI (the reference-free version)², we follow Vernikos et al. (2022) and add aligned and concatenated context as inputs to the model-based metrics. We call these metrics CW-COMET and CW-KIWI. We use the same context size of 3 as described in Section 3.3.

Sentence Segmentation and Alignment. Similar to IWSLT evaluation paradigm for speech translation (Ahmad et al., 2024), we propose to treat sentence segmentation and alignment as part of the evaluation for our paragraph-level evaluation. We follow the same general procedure as described in Section 3.3. We handle sentence alignment differently according to whether the metric requires reference input. Misaligned sentences remain in their original positions, with empty strings as placeholders where necessary. If a sliding window contains only empty strings on the one side, a default score of zero is given. This penalizes over-translation and omission, aligning with our research objectives. We verified the validity of our paradigm via a sanity check. Details are in Appendix D.

¹Unbabel/wmt22-comet-da

²Unbabel/wmt22-cometkiwi-da

In addition to reporting model-based metrics, because over- and under-translations usually create significant misalignments across sentences, we track them by reporting the ratio of 1-0 and 0-1 alignments (1-0/0-1 Ratio), which represents the proportion of windows in the source or translation that fail to align. This is evaluated on our sentence-segmented and sentence-aligned test set. We also report document-level BLEU (d-BLEU) as some previous work has done, even it does not correlate well with human judgments (Deutsch et al., 2023) and should only be treated as an auxiliary metric.

4.4 Baselines

We evaluate all training-time alignment methods on LLaMA-3.1-8B-Instruct and also use it as the backbone model for test-time alignment methods, including Plan2Align. This ensures consistency in comparing the effectiveness and aligns with our focus on small-scale LLMs. Additionally, we compare these results against the original LLaMA-3.1-8B-Instruct to further assess improvements and changes, serving as the baseline for performance. Implementation details of Plan2Align, including parameters and prompt design, are in Appendix C.

GPT-40 and Qwen-2.5-14B. We select GPT-40 as the upper bound for performance. GPT-40 is not optimized for translation, but closely approaches that of specialized translation systems (Shahriar et al., 2024). Its precedent model, GPT-4, was the top-ranked model for 5 langauge pairs in WMT24 general machine translation shared task (Kocmi et al., 2024). In addition, we also include Qwen-2.5-14B for comparison, as it demonstrates strong performance in Chinese language contexts.

Test-Time Alignment Models. We select ARGS (Khanov et al., 2024) and RAIN (Li et al., 2024) as they both represent test-time alignment methods similar to Plan2Align but with distinct operational mechanisms. ARGS is selected for its reward-guiding method similar to Plan2Align's; however, while Plan2Align evaluates the entire context with a reward model, ARGS dynamically recalculates scores for each token during generation. On the other hand, RAIN is selected for its ability to score newly generated tokens and perform auto-regressive self-evaluation, requiring a start-from-scratch search and backward progression. To better observe the benefits brought by MPC, we also compare it with Best-of-N and Vanilla MPC. The

implementation details of test-time alignments, are available in Appendix B.1.

Training-Time Alignment Models. We also compare Plan2Align with various training-time alignment methods. We consider SFT directly on the same preference dataset. Moreover, we include SimPO for comparison as it serves as a strong benchmark for training-time alignment in general. We also consider DPO as it serves as a mainstream representative of the training-time-alignment series, with training details available in Appendix B.2.

4.5 Quantitative Results

We present the primary results for **zh**→**xx** in Table 1. Our primary goal is not to outperform all models but rather to maximize the utility of existing models, approaching or even surpassing training-time alignment methods. Since training-time methods have direct exposure to real paragraph data, direct comparisons to test-time methods would be unfair. Instead, we emphasize whether Plan2Align can consistently improve existing model performance. We summarize the observations as follows:

Plan2Align Outperforms Existing Test-Time Alignment Methods. Table 1 demonstrates that Plan2Align outperforms test-time alignment baselines across all translation directions. This is primarily because paragraph-level MT requires generating long outputs, and existing methods such as ARGS and RAIN guide the decoding process at the token level through a reward model. These methods struggle with error accumulation, where poor early-context translations negatively affect subsequent generations. Additionally, Table 2 highlights the efficiency of Plan2Align compared to ARGS and RAIN, detailing the time required for each approach across iterations. Since Plan2Align operates directly at the context level, it generates complete translated paragraphs without additional decoding overhead, making it both faster and more effective.

Plan2Align Closes the Performance Gap Between Test-Time and Training-Time Alignment Methods. For zh→en, Plan2Align significantly enhances LLaMA-3.1's performance, surpassing training-time methods in CW-COMET, demonstrating that test-time alignment can effectively bridge the gap with training-time approaches. For zh→ru, while CW-COMET achieves the best performance, CW-KIWI lags behind training-time methods, likely due to KIWI-22's limitations in this

language pair, preventing it from reflecting CW-COMET's improvements. Notably, Plan2Align surpasses Qwen in CW-COMET, highlighting its effectiveness in low-resource settings such as Russian, where Qwen may not be fully optimized. Despite LLaMA-3.1's limited Russian proficiency, Plan2Align's self-rewriting and planning-based alignment mitigate these weaknesses, improving translation quality. For **zh**→**de**, while Plan2Align does not surpass training-time and SoTA models, it exhibits only a small performance gap. Crucially, Plan2Align is model-agnostic and seamlessly integrates with existing architectures. Applying it to the strongest training-time model (SFT) $(i.e., Llama-3.1_{Plan2Align_{SFT}})$, further enhances performance, bringing it closer to SoTA levels.

Best-of-N, Vanilla MPC, and Plan2Align. discussed in Section 1, non-extensive LLMs struggle to generate high-quality paragraph translations in a single pass. This limitation renders Bestof-N ineffective for paragraph-level MT, while directly applying MPC (i.e., Vanilla MPC) fails due to the model's difficulty in selecting optimal contexts for subsequent translations. Table 1 shows that Best-of-N improves translation quality but does not enhance the model's fundamental capabilities. Its effectiveness remains limited by randomness, making it an inefficient approach. Even with N=60 (six times the number of iterations in Plan2Align), it still underperforms compared to Plan2Align across all translation directions. Vanilla MPC achieves improvements for zh-en, but its effectiveness diminishes in $\mathbf{zh} \rightarrow \mathbf{ru}$ and $\mathbf{zh} \rightarrow \mathbf{de}$, sometimes even falling below LLaMA-3.1's baseline performance. This suggests that for less familiar languages, LLMs struggle to extract meaningful improvements. Plan2Align overcomes this limitation through context alignment and self-rewriting, enabling the effective application of MPC to paragraph-level MT.

Impact of Iteration Number in MPC. As iterations increase, the context buffer expands, and Plan2Align selects more reference contexts. However, over-translation and omission often prevent perfect semantic alignment. Using excessive reference contexts in the self-rewriting task can overload the LLM, making it harder to extract meaningful insights due to overlapping semantics. To isolate translation difficulty effects, we focus on $zh \rightarrow en$ performance. Table 3 shows that while increasing reference contexts improves translation initially,

Methods	AVG. Time Consumed				
Methous	zh→en	zh→de	zh→ru		
ARGS	1686	2343	2035		
RAIN	4639	5518	4454		
Plan2Align	311	313	424		

Table 2: Average execution time (in seconds) per paragraph for three test-time alignment methods on the validation set. Plan2Align, with an iteration number of 3, is significantly faster than ARGS and RAIN since it avoids token-level alignment. Tests are conducted on a single NVIDIA RTX 6000 Ada Generation GPU.

	Iteration	CW-COMET ↑	CW-KIWI↑	d-BLEU↑	1-0/0-1 Ratio ↓
	1	93.28	66.89	20.99	0.32
. <u>ig</u>	2	93.82	66.72	20.75	0.52
ZA	3	94.13	67.06	21.17	0.20
Plan2Align	4	93.95	66.19	20.81	0.38
	5	93.97	66.59	21.09	0.35
C	1	89.79	58.40	17.30	4.55
MPC	2	88.00	56.26	16.45	6.42
a	3	83.92	53.08	15.11	10.70
Vanilla	4	84.66	53.60	15.26	9.93
\Za_	5	84.93	54.03	15.48	9.68

Table 3: Performance of Plan2Align and Vanilla MPC across different iterations on the $zh \rightarrow en$ task.

excessive references (iteration = 4) degrade performance due to semantic ambiguity. Interestingly, Vanilla MPC exhibits an even sharper decline that is starting at iteration = 2, even in the simpler $zh \rightarrow en$ task. This supports our hypothesis that LLMs cannot directly leverage MPC effectively. Small-scale models struggle to extract useful contexts, leading to error accumulation and deteriorating translations. Plan2Align overcomes this limitation through structured context alignment and self-rewriting, ensuring stable improvements in paragraph-level MT.

5 Conclusion

In this paper, we introduced Plan2Align, a new form of model predictive planning framework for test-time preference alignment in paragraph-level machine translation. By recasting translation as a trajectory optimization problem and leveraging a self-rewriting mechanism, our approach effectively addresses key challenges such as semantic drift, omissions, and incoherence that arise in onepass, next-token generation. Our results suggest that predictive planning can robustly accumulate and utilize high-quality translation contexts, paving the way for further improvements in long-form machine translation. Future work will explore broader language pairs and the integration of training-time strategies to further enhance translation quality. Our code will be open source under MIT license.

Limitation

One of our limitation lies in the scope of translation languages. Our experiments have focused on language pairs (e.g., Chinese to English, German, and Russian) from reproducible benchmarks, and while these choices provide valuable insights, they could potentially not encompass the full linguistic diversity present in the other translation tasks. Languages with complex morphologies, low-resource languages, or those with radically different syntactic structures may present challenges that our current formulation of context alignment and iterative refinement does not fully address. Future work has been scheduled to explore a broader set of languages to ensure that the method generalizes well and does not inadvertently favor certain language families or linguistic features over others.

Ethical and Societal Considerations While Plan2Align is designed to enhance translation coherence and quality, its deployment raises important ethical and societal considerations. First, there is an inherent risk of bias amplification. The reward model, which drives context selection during test-time alignment, is trained on preference data that may itself contain implicit biases. If not carefully audited and diversified, this could lead to translations that systematically favor dominant linguistic norms or cultural perspectives, marginalizing underrepresented human dialects. In sum, by focusing on a limited set of languages and model architectures, there is a risk of reinforcing existing digital divides. If high-quality translation tools are predominantly developed for well-resourced languages and large-scale models, speakers of lowresource or underrepresented languages may be left behind. As researchers and practitioners, it is a long-term developing responsibility to prioritize inclusivity and fairness, ensuring that technological advances in machine translation benefit a diverse research community.

Acknowledgment This research has been supported by NVIDIA grant and technical involvement from NVIDIA Taiwan AI R&D Center (TRDC).

References

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

Ibrahim Said Ahmad, Antonios Anastasopoulos, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, William Chen, Qianqian Dong, Marcello Federico, Barry Haddow, Dávid Javorský, Mateusz Krubiński, Tsz Kin Lam, Xutai Ma, Prashant Mathur, Evgeny Matusov, Chandresh Maurya, John McCrae, Kenton Murray, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Atul Kr. Ojha, John Ortega, Sara Papi, Peter Polák, Adam Pospíšil, Pavel Pecina, Elizabeth Salesky, Nivedita Sethiya, Balaram Sarkar, Jiatong Shi, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Alex Waibel, Shinji Watanabe, Patrick Wilken, Petr Zemánek, and Rodolfo Zevallos. 2024. Findings of the IWSLT 2024 Evaluation Campaign. In Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024), pages 1–11, Bangkok, Thailand (in-person and online). Association for Computational Linguis-

Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. Tower: An Open Multilingual Large Language Model for Translation-Related Tasks. In First Conference on Language Modeling.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-shot Learners. *Advances in neural information processing systems*, 33:1877–1901.

E. F. Camacho and C. Bordons. 2007. *Constrained Model Predictive Control*, pages 177–216. Springer London, London.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. *Advances in neural information processing systems*, 31.

Daniel Deutsch, Juraj Juraska, Mara Finkelstein, and Markus Freitag. 2023. Training and Meta-Evaluating Machine Translation Evaluation Metrics at the Paragraph Level. In *Proceedings of the Eighth Conference on Machine Translation*, pages 996–1013, Singapore. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond English-Centric Multilingual Machine Translation. *J. Mach. Learn. Res.*

Zhaopeng Feng, Yan Zhang, Hao Li, Bei Wu, Jiayu Liao, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2024. Tear: Improving LLM-based Machine Translation with Systematic Self-refinement. *Preprint*.

- Patrick Fernandes, Kayo Yin, Emmy Liu, André Martins, and Graham Neubig. 2023. When Does Translation Require Context? A Data-driven, Multilingual Exploration. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), pages 606–626, Toronto, Canada. Association for Computational Linguistics.
- Mohamed Seghir Hadj Ameur, Ahmed Guessoum, and Farid Meziane. 2019. Improving Arabic Neural Machine Translation via N-best List Re-ranking. *machine translation*, 33:279–314.
- Nicklas A Hansen, Hao Su, and Xiaolong Wang. 2022. Temporal Difference Learning for Model Predictive Control. In *International Conference on Machine Learning*, pages 8387–8406.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou,
 Mantas Mazeika, Dawn Song, and Jacob Steinhardt.
 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Yuchen Hu, Chen Chen, Chao-Han Yang, Ruizhe Li, Dong Zhang, Zhehuai Chen, and EngSiong Chng. 2024. GenTranslate: Large Language Models are Generative Multilingual Speech and Machine Translators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 74–90, Bangkok, Thailand. Association for Computational Linguistics.
- Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. 2016. Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints. *IEEE Transactions* on Vehicular Technology, 66(2):952–964.
- Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google Submission to the WMT 2024 Metrics Shared Task.
 In Proceedings of the Ninth Conference on Machine Translation, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.
- Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. ARGS: Alignment as Reward-Guided Search. In *The Twelfth International Conference on Learning Representations*.
- Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof

- Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.
- Basil Kouvaritakis and Mark Cannon. 2016. Model Predictive Control. *Switzerland: Springer International Publishing*, 38:13–56.
- Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. 2024. RAIN: Your Language Models Can Align Themselves without Finetuning. In *The Twelfth International Conference on Learning Representations*.
- Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The Unlocking Spell on Base LLMs: Rethinking Alignment via In-Context Learning. In *The Twelfth International Conference on Learning Representations*.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
- Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. 2019. Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. In *International Conference on Learning Representations*.
- Angrosh Mandya, James O' Neill, Danushka Bollegala, and Frans Coenen. 2020. Do Not Let the History Haunt You: Mitigating Compounding Errors in Conversational Question Answering. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2017–2025, Marseille, France. European Language Resources Association.
- Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2022. A Survey on Document-level Neural Machine Translation: Methods and Evaluation. *ACM Computing Surveys*, 54(2):45:1–45:36.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple Preference Optimization With a Reference-Free Reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. 2020. Deep Dynamics Models for Learning Dexterous Manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR.
- OpenAI. 2024. Hello GPT-40 (API version 40-0211). Accessed: 2025-02-14.
- Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*.
- Matt Post and Marcin Junczys-Dowmunt. 2024. Evaluation and Large-Scale Training for Contextual Machine Translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1125–1139, Miami, Florida, USA. Association for Computational Linguistics.
- Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. 2024. TreeBoN: Enhancing Inference-Time Alignment with Speculative Tree-Search and Best-of-N Sampling. Computing Research Repository.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Anil V. Rao. 2014. *Trajectory Optimization: A Survey*, pages 3–21. Springer International Publishing, Cham.
- Sakib Shahriar, Brady D. Lund, Nishith Reddy Mannuru, Muhammad Arbab Arshad, Kadhim Hayawi, Ravi Varma Kumar Bevara, Aashrith Mannuru, and Laiba Batool. 2024. Putting GPT-40 to the Sword: A Comprehensive Evaluation of Language, Vision, Speech, and Multimodal Proficiency. *Computing Research Repository*, abs/2407.09519.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Kyle Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, et al. 2023. Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models. Transactions on Machine Learning Research. Featured Certification.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to Summarize with Human Feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

- Yuval Tassa, Tom Erez, and Emanuel Todorov. 2012. Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4906–4913. IEEE.
- Yuval Tassa, Nicolas Mansard, and Emo Todorov. 2014. Control-limited Differential Dynamic Programming. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 1168–1175. IEEE.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv preprint arXiv:2408.00118.
- Brian Thompson and Philipp Koehn. 2020. Exploiting Sentence Order in Document Alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007, Online. Association for Computational Linguistics.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. LLaMA: Open and Efficient Foundation Language Models. *arXiv* preprint *arXiv*:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open Foundation and Fine-tuned Chat Models. *arXiv preprint arXiv:2307.09288*.
- Giorgos Vernikos, Brian Thompson, Prashant Mathur, and Marcello Federico. 2022. Embarrassingly Easy Document-Level MT Metrics: How to Convert Any Pretrained Metric into a Document-Level Metric. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 118–128, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.
- 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, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-Level Machine Translation with Large Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Mozhi Zhang, Ke Ren, Botian Jiang, and Xipeng Qiu. 2024b. InferAligner: Inference-Time Alignment for Harmlessness through Cross-Model Guidance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10460–10479, Miami, Florida, USA. Association for Computational Linguistics.
- Minghao Wu, George Foster, Lizhen Qu, and Gholamreza Haffari. 2023. Document Flattening: Beyond Concatenating Context for Document-Level Neural Machine Translation. In *Proceedings of the 17th* Conference of the European Chapter of the Association for Computational Linguistics, pages 448–462, Dubrovnik, Croatia. Association for Computational Linguistics.
- Qiyu Wu, Masaaki Nagata, Zhongtao Miao, and Yoshimasa Tsuruoka. 2024. Word Alignment as Preference for Machine Translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3223–3239, Miami, Florida, USA. 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.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Kaifeng Yun, Linlu GONG, Nianyi Lin, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Xu Bin, Jie Tang, and Juanzi Li. 2024. KoLA: Carefully Benchmarking World Knowledge of Large Language

- Models. In *The Twelfth International Conference on Learning Representations*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. 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)*.
- Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314, Mexico City, Mexico. Association for Computational Linguistics.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-Tuning Language Models from Human Preferences. *arXiv* preprint arXiv:1909.08593.

A Details of Preference Data

We used MetricX-24-XL to labeled our pairwise preference datasets, the Chinese sources sentences are from WMT'24 Discourse-Level Literary Translation benchmark, we employed LLaMA-3.1-8B-Instruct, Gemma-2-9B, and GPT-4o to generate translations across three language pairs; each language pair has 2000 records, with a maximum length of 1024 tokens. The distribution of preferences, indicating the number of translation is the best translation among the three, indicating the number of translation is the best translation among the three, given that MetricX-24 scores range from 0 to 25, where 0 is the best and 25 is the worst. We removed translations scoring above 20, and if two out of three translations in a paragraph exceeded this threshold, we did not use that paragraph. The details are in Table 4. Our test-time setting, pretrained models, and preference data will be open source under MIT license.

	LLaMA Wins	Gemma Wins	GPT4 Wins
zh→en	310	421	1269
$zh{ o}de$	82	99	1814
zh→ru	32	127	1680

Table 4: The statistics of winning translations for each language pairs evaluated by MetricX-24.

B Implementation Details of Baselines

B.1 Test-Time Alignment Methods

B.1.1 ARGS

We follow the setting of ARGS, We adopt the ARGS-greedy method in ARGS as our baseline. Following the setting of ARGS-greedy. We set w to 1.5 and k to 10. For fairness, we replaced the backbone model with the same LLaMA3.1-8B-Instruct as Plan2Align, and the reward model was also replaced with the reward model used by Plan2Align. Although ARGS settings indicate that using ARGSgreedy results in answers more closely aligned with the characteristics specified in the reward model, ARGS uses the weighted sum of logit of the token and the reward for token generation. Given that the number of tokens generated by ARGS-greedy does not exceed those produced by Plan2Align and RAIN, we included ARGS-stochastic for comparison and conducted best-of-n to optimize results, the choice of n was determined based on the average number of tokens required by Plan2Align and RAIN to generate a single translation.

However, ARGS-stochastic's best-of-n did not surpass ARGS-greedy in performance, leading us to ultimately select ARGS-greedy as the baseline.

B.1.2 RAIN

In RAIN, we also replaced the backbone model with LLaMA3.1-8B-Instruct and replaced the self-evaluation prompt in RAIN with text in the Figure 3. For parameters in RAIN, we set value threshold V to 0.8 as the default setting of RAIN. We try four combinations of maximum and minimum number of search iterations for finding the parameter that generates the required number of tokens close to Plan2Align, which are [10,20]. The detailed configuration of the tokens generated in each parameter pairs is in Table 5.

(MinT,MaxT)	AVG. tokens
[6,12]	6575
[8,16]	8972
[10,20]	13401
[12,24]	17291

Table 5: The average number of tokens required to generate a translation for RAIN in each maximum and minimum number of search iterations pairs.

B.1.3 Vanilla MPC

Vanilla MPC applies MPC directly to paragraph-level machine translation (MT) by selecting the best output from the previous iteration and refining it. We conduct six iterations, with performance results presented in Table 6. The best performance for each language pair (iteration = 1, 2, 4 for zh \rightarrow en, zh \rightarrow ru, and zh \rightarrow de, respectively) is reported in Table 1. The results show that even in the zh \rightarrow en translation direction, performance continues to decline over iterations. For more challenging language pairs (zh \rightarrow ru and zh \rightarrow de), LLMs struggle to extract useful contexts, preventing sustained improvement and leading to unstable performance.

B.2 Training-Time Alignment Methods

We choose LLaMA3.1-8B-Instruct as our experiments' backbone model, including the reward model and training-time methods. We utilize one NVIDIA RTX 6000 Ada Generation GPU to train models with the LLaMA Factory library³ (Zheng et al., 2024). For the training setups, the SFT model is trained on preferred data from the preference dataset, while the Reward Model, DPO,

³https://github.com/hiyouga/LLaMA-Factory

	Iteration	$\text{CW-COMET} \uparrow$	CW-KIWI↑	d-BLEU \uparrow	1-0/0-1 Ratio ↓
	1	89.79	58.40	17.30	4.55
	2	88.00	56.26	16.45	6.42
zh-en	3	83.92	53.08	15.11	10.70
4	4	84.66	53.60	15.26	9.93
	5	84.93	54.03	15.48	9.68
	6	84.48	52.91	15.12	10.08
	1	73.80	30.83	5.54	20.00
	2	82.83	35.18	7.19	10.26
e	3	77.08	34.08	6.15	16.40
zh-en	4	79.56	33.25	6.21	13.74
.,	5	78.35	33.31	6.34	15.35
	6	78.35	33.37	5.43	15.24
	1	83.06	39.00	10.89	11.19
	2	80.88	36.74	8.95	13.03
e	3	79.79	36.72	9.70	14.51
zh-en	4	83.67	38.46	10.82	10.63
	5	82.72	38.54	10.25	11.40
	6	79.35	35.87	9.54	14.87

Table 6: Performance of Vanilla MPC across different iterations.

and SimPO are trained on the full preference data that includes inputs, good translations, and bad translations. All models are configured with an identical set of hyperparameters: they utilize the AdamW(Loshchilov and Hutter, 2019) optimizer and feature gradient accumulation steps fixed at 8, with a cutoff length of 2048. The maximum allowable gradient norm is restricted to 1.0 to ensure training stability. All training procedures are conducted using bf16 precision, each model undergoes a single epoch of training with a batch size of 2, and the LoRA(Hu et al., 2022) rank and LoRA alpha parameters are set to 16 across all models. The learning rates for the models are: Rewarg Model at 1e-5, SFT at 2e-5, and both DPO and SimPO at 1e-6 as suggested by (Rafailov et al., 2023) and (Meng et al., 2024). Additionally, SimPO utilizes a beta and gamma of 2.5, which aligns with its setup, whereas SFT and DPO use a beta of 0.1 also followed by (Rafailov et al., 2023). When Plan2Align serves as a generic method, we also plan to connect the method for multi-agent (i.e., cascaded) based translation system (Hu et al., 2024) in the future.

C Implementation Details of Plan2Align

In this section, we describe our parameter settings and prompt design, as well as the impact of different prompts on the final translation generation. Our full approach is outlined in Algorithm 1.

C.1 Parameter Setting

To generate diverse translations for comparison, we design three distinct system prompts for the self-rewriting task, resulting in three translation outputs per iteration. The context alignment win-

dow size is set to 3, the context buffer threshold is fixed at 2, and the iteration number is set to 3. When constructing the self-rewriting task input from the context buffer, we employ a sampling strategy to enhance translation diversity and identify more valuable contexts: As the number of iterations increases, the content within the context buffer expands. Consequently, we adaptively increase the number of reference contexts selected by Plan2Align. Specifically, we tie the number of reference contexts to the current iteration count using the formula: $reference\ context\ number = current\ iteration\ number + 5$.

C.2 Prompt Design

Here is the prompt configuration for generating translations in Plan2Align: Figure 4 displays our prompt template for the self-rewriting task within Plan2Align. We employ three distinct system prompts that yield three types of translations: sentence-by-sentence translation, precise translation, and imaginative translation. These prompts cater to different translation needs within the context buffer, allowing us to extract the most effective parts. Additionally, we use a context prompt to ensure the quality of the translations produced. Figures 5 and 6 showcase the prompts used for generating the final translations in Plan2Align. The distinctions between these two prompts, Reference and Annotation, are further explored in Section C.2.1.

C.2.1 Impact of Final Translation Task

In Plan2Align, final translation generation differs from context buffer updates. The final translation is produced by assembling buffer-stored contexts in source sentence order, forming a reference translation (Reference). Conversely, during context buffer updates, translated sentences are annotated beneath their corresponding source sentences (Annotation). Table 7 shows that the Reference approach outperforms Annotation in **zh**→**ru** and $\mathbf{zh} \rightarrow \mathbf{de}$. However, for $\mathbf{zh} \rightarrow \mathbf{en}$, the Annotation method generally performs better. This difference arises because annotating each source sentence with context buffer information and refining segments independently, although not requiring explicit alignment, demands higher semantic understanding. This makes it effective for **zh** \rightarrow **en** but burdensome for the other two translation directions. In contrast, the Reference approach offers greater flexibility in refining the entire paragraph.

Algorithm 1 Plan2Align

```
Inputs: Source x, number of MPC iterations N, base model LM, reward model r
Output: Translation result y_{final}
1: Initial buffer B
2: for k = 0 \to N - 2 do
3:
       y \leftarrow LM(x, prompt in Figure 4)
                                                                                                  // Translation from 3 system prompt
       (W_x, W_y) \leftarrow Make source and translation into aligned context windows
                                                                                                                    // See Section 3.1.1
5:
       for all (w_x, w_y) \in (W_x, W_y) do
           reward \leftarrow r(w_x, w_y)
6:
                                                                                          // Compute rewards for each window pair
           Update the best window at each position to put into Buffer {\cal B}
7:
                                                                                                                   // Window selection
8:
9:
       y_{final} \leftarrow \text{LM}(\text{sentences in } B, \text{ prompts in Figures 5})
10: end for
11: return y_{final}
```

	Prompt Type	CW-COMET ↑	CW-KIWI↑	d-BLEU↑	1-0/0-1 Ratio ↓
zh-en	Annotation Reference	94.02 94.13	67.83 67.06	22.16 21.17	0.38 0.20
zh-ru	Annotation	80.58	35.42	10.09	12.83
	Reference	91.23	40.11	10.69	1.49
zh-de	Annotation	80.25	37.27	9.09	13.54
	Reference	92.60	44.62	10.69	1.44

Table 7: Performance of Plan2Align across two different final translation tasks.

D Evaluation Sanity Check

To verify the validity of our sentence segmentation and alignment approach in the context of machine translation evaluation, we conduct an evaluation using the test set for WMT24 general machine translation translation shared task, as it is a sentence-aligned dataset. We select Zh-En and En-Zh translation data for our sanity check. Specifically, we concatenate sentences with spaces to simulate paragraph structures while ensuring that each paragraph does not exceed 1024 tokens, resulting in a total of 1181 paragraphs. In addition, we simulate imperfect alignments with under- and over-translations by creating two extra scenarios: (1) randomly dropping 10% of source sentences and (2) randomly dropping 10% of target sentences.

For simplicity, we do not generate system translation and directly evaluate the source-reference pair using KIWI-22. For our evaluation paradigm to be valid, the score distribution should change similarly when over- and under-translations are introduced. We also track the change in sentence pairs in our sanity check, but we expect that to change with dropped sentences because one-to-many and many-to-one alignments may be generated.

Table 8 presents the alignment performance under different conditions. It can be seen that our score distribution reacts similarly to the score calculated with the ground-truth alignment when underand over-translation errors are introduced. This

demonstrates that our evaluation method effectively captures semantic alignment and validates the applicability of CW-COMET and CW-KIWI in reflecting context-level translation quality.

Scenario	# Pairs	Mean	Std.
Perfectly Aligned	22956	78.03	7.77
Our Alignment	22310	77.07	9.79
Drop 10% (Source)	22956	73.68	15.20
Our Alignment	21222	73.81	14.20
Drop 10% (Target)	22956	75.25	11.78
Our Alignment	21433	74.97	11.94

Table 8: Alignment performance across different conditions. The change in our score maintains high consistency with the score calculated with perfect alignment.

E MetricX Scoring Failure Cases

Here are three failure cases for metricX-24, each illustrated with different models and language pairs:

RAIN Model, zh-ru Pair. The translation output appears nonsensical (see Figure 7), yet metricX-24 awarded it a score of 8.76, indicating a seemingly better translation under a metric where lower scores denote superior results.

ARGS Model, zh-de Pair. As shown in Figure 8, ARGS produced repetitive phrases without completing the translation from the original zh-de text, reaching the max token limit. Despite this, it received a score of 8.47 from metricX-24.

LLaMA3.1 Model, zh-en Pair. As shown in Figure 9, the translation did not start from the beginning of the text, and the meaning conveyed was incomplete. Nevertheless, metricX-24 gave it a high score of 9.85, which inaccurately suggests a less effective translation.

```
[INST]
Consider the following source text (Source) and its translation (Translation).
Determine if the translation is accurate.
Translations that deviate from the objective meaning of the source text, introduce speculative content, or alter the intended meaning are considered inaccurate.
<generated text>
Options:
(A) The translation is accurate.
(B) The translation is inaccurate.[/INST]
The evaluation is: (
```

```
[INST]
Consider the following source text (Source) and its translation (Translation).
Determine if the translation is accurate.
Translations that deviate from the objective meaning of the source text, introduce speculative content, or alter the intended meaning are considered inaccurate.
<generated text>
Options:
(A) The translation is inaccurate.
(B) The translation is accurate.[/INST]
The evaluation is: (
```

Figure 3: Prompt templates for RAIN.

```
processed_source = [
·
相反,眼前的這只手臂纖細瘦弱,因為常年沒有照射到太陽的緣故,皮膚有些病態的蒼白。
這是現實中自己的身體,他很清楚這一點。
但是, 自己怎麼會受了傷的?
而且, 這里也不象是醫院啊?
羅德擡頭望去,整個房間看起來好像是個艙室,沒有燈,也沒有電話,更沒有呼叫鈴。
一張木桌, 兩把椅子以及一個
(A blonde-haired girl in a white robe walked in, her eyes wide with surprise as she gazed at
Rod, who was half-sitting up. )
固定在墻邊的櫃子就是這里的全部家當。
不知道為什麼,羅德覺得自己似乎在什麼地方見到過這個場景似的。
(For some reason, Rod felt that he had seen this scene before, as if it were familiar to him. )
而就在羅德仔細打量這個房間時, 門忽然打開了。
(Just as Rod was scrutinizing the room, the door suddenly swung open. )
system_prompts = [
"You are a meticulous translator. Provide a literal, word-for-word translation that
preserves the structure and meaning of each individual word.",
"You are a professional translator. Deliver a clear, formal, and precise translation that
faithfully conveys the original meaning.",
"You are a creative and expressive translator. Render the text in a vivid and imaginative
way, as if narrating a captivating story."
context_prompt =
f"Below is a specialized, intermediate translation task. The input text is a mix of Chinese and
preliminary { language} translations enclosed in parentheses. These provided translations are rough
references - they may be incomplete, inconsistent, or not fully aligned with the original
meaning. Your task is to produce an improved <code>language</code>} translation according to the following
guidelines:
1. **Refinement:** For sections with existing <code>language</code>} translations (in parentheses), refine and
polish them so that they are fluent, accurate, and coherent, fully capturing the meaning of the
corresponding Chinese text.
2. **Completion:** For sections that remain untranslated, translate the Chinese text accurately
and naturally in the specified style.
3. **Translation Order and Structure Preservation: ** Maintain the original order and structure of
the text. Every Chinese sentence must appear in the same sequence as in the source text, with its
corresponding { language } translation (if available) inserted immediately after it. Do not
rearrange or reorder any part of the text.
4. **Consistency:** Ensure a uniform tone and style across the entire translation, adhering to
the translator role specified.
5. **Final Output:** Provide the final output as a single, well-structured <code>language</code>} text. Do not
include any extraneous commentary, explanations, annotations, or headers - output only the
translation in the correct order.
Note: This translation is an intermediate version that may later be merged with other
translations. Focus on clarity, coherence, and fidelity to the source text.
Here is the input data for translation:\nprocessed\_source\n
Apply the above quidelines to produce an improved, coherent translation that strictly follows the
original order of the text.
input_messages = [
                {"role": "system", "content": prompt},
                {"role": "user", "content": context_prompt}
            1
```

Figure 4: Prompt template used for self-rewriting task in Plan2Align and an actual example.

```
相反,眼前的這只手臂纖細瘦弱,因為常年沒有照射到太陽的緣故,皮膚有些病態的蒼白。
這是現實中自己的身體,他很清楚這一點。
但是,自己怎麼會受了傷的?
而且, 這里也不象是醫院啊?
羅德擡頭望去,整個房間看起來好像是個艙室,沒有燈,也沒有電話,更沒有呼叫鈴。
一張木桌,兩把椅子以及一個固定在墻邊的櫃子就是這里的全部家當。
不知道為什麽,羅德覺得自己似乎在什麽地方見到過這個場景似的。
而就在羅德仔細打量這個房間時, 門忽然打開了。
initial_translation = [
Contrary to this, the arm that was right in front of him was slender and weak, pale-skinned due
to a lack of sunlight for a long time.
This was his own body in the real world, a fact he was well aware of.
But how did he end up injured?
And besides, this doesn't look like a hospital at all!
Rod lifted his head to take in the room, which resembled a cramped compartment. There were no
lights, no phones, and no alarm bells.
There was only a wooden table, two chairs, and a cabinet fixed to the wall.
For some reason, Rod felt that he had seen this scene before, as if it were familiar to him.
Just as Rod was scrutinizing the room, the door suddenly swung open.
rewrite_prompt =
f"Below is an initial translation of a Chinese text into {language}. This translation may
include omissions, inaccuracies, or awkward phrasing. Your task is to produce a refined
version that is fluent, accurate, and coherent, while faithfully preserving the full
meaning of the original Chinese text.\n\"
### Instructions:\n
1. Ensure that every detail in the original Chinese text is accurately represented.\n
2. Correct any grammatical errors, unnatural expressions, or inconsistencies.\n
3. Improve the natural flow so that the translation reads as if written by a native
speaker.\n
4. Do not add, omit, or change any essential details from the source text. \n
5. Output only the final refined translation without any additional commentary.\n
### Original Chinese Text:\n{source_sentence}\n\n"
### Initial { language } Translation: \n{initial_translation} \n\n"
### Refined Translation:"
```

source sentence =

Figure 5: Prompt used for generating final translation in Plan2Align (the *Reference* version of the prompt in Section C.2.1) and an actual example.

Figure 6: Applying the concept of the self-rewriting task to generate the final translation prompt (the *Annotation* version of the prompt in Section C.2.1).

< SRC

而三皇女......她總是一副急色猥瑣的模樣,破壞了原本的美感,這會兒臉上沒有什麽表情,那份美頓時顯現出來了,眼尾輕挑,艷 色幾乎令人不敢直視。 少年看得微微呆住,那句誇讚不由自主便 說出了口。如果放在其他貴女那里,這算得上逾距了,但對喜歡調戲男子的三皇女來 說, 卻是情趣。 三皇女瞥了他一眼,果然沒訓斥他有失規矩。少年心里一喜,動作更殷勤了一些,小心地挽好皇女的長發。 只是,直到梳妝快結束,今天一直隱隱期待著的調戲,也並沒有到來。少年咬了咬嘴唇,小心試探道: "殿下. ...可是心情不好?" 身為男子,還是侍從,為自己的貴女排憂解難,溫柔安撫,是件很正常的事。以往的三皇女,也總是喜歡這樣溫柔小意的伺候。 但今天顯然不一樣。三皇女沒有露出一絲心情不好的表情,看起來和往常一樣,但偏偏對他冷淡了許多 "沒有。"連回答都如此簡單。 少年咬唇,仔細想了想,小聲安慰道,"殿下勿需擔心,彌心郎君只是心氣高些,他早晚會知道殿下的好的。 " 喻楚挑了挑眉。葉彌心 ──是丞相府的公子。也是劇情中那個所謂的美少年。 身份的尊貴養出他 絕好的氣度,容貌又俊美,身形修長好看。 京都不知道多少女人暗暗垂涎他,卻礙於身份,搶不能搶,勾搭吧,人家又心氣高,至今沒人能勾搭到手 三皇女也是垂涎大軍中的一員。雖 說不敢直接逼丞相的公子嫁她,但她仗著身份,總是沒臉沒皮地實施勾搭大計。結果也可想

< MT >

而知。

Figure 7: Case 1: Failed evaluation on **RAIN**, zh-ru, metricX score: 8.76, produced nonsensical output.

< SRC >

喬修說出了她的名字,這個角色可以 說是貫穿了魔獸世界整個故事的女主角,同樣也是《爐石傳 說》這個遊戲 `法師'這個職業的代表。

"她和你是什麼關系…"希里稍微沈默了一會,還是沒忍住向喬修問出了這個問題。

"關系?"喬修在第一時間楞了一下,吉安娜只是遊戲中的一個虛擬人物,喬修和她並沒有任何的關系,但作為一個魔獸玩家,喬修覺得這麼回答也不太對。

"我想應該是曾經一起作戰的同伴。"

喬修當初玩魔獸世界玩的是聯盟陣營,和吉安娜同處一個陣營,以遊戲玩家這一身份而言,喬修確實算得上是和吉安娜一起作戰過 的同伴。

希里沒說話,她看了一眼吉安娜的肖像畫,突然又想起了之前在城堡里面睡覺時,偶爾 偷聽到的喬修和澤那斯的對話,她記得喬修 曾說了一句`我已經有喜歡的人了'。

喬修能猜得到希里在想什麽,但喬修沒有刻意的去解釋,只是把視線看向了吉安娜的肖像畫上。

魔導機械中投影的畫作是希里根據喬修的原稿臨摹的,希里所畫出來的效果再次讓喬修意識到了希里壓根就不適合當一個魔法師!

喬修只是簡單告訴了希里一些上色和構圖的技巧,就讓希里照著他給的原稿在魔導機械上重新繪制。

最後希里畫出來的結果雖然比起、專業,還有一定距離,但起碼已經步入了、還能用,的範疇。

最重要的一點是希里的作畫沒有受到這個時代的影響,這個時代的繪畫都是以油畫為主,畫作的畫風更加貼近《蒙娜麗莎》的微笑 那種感覺,還有中 -國的水墨畫風格,這種風格的作畫,喬修曾在白荊花爵士的劇院中看見過,聽白荊花爵士 說,這幅畫的畫家來 自於一個名為 `大夏'的國家。

但無論是油畫還是水墨畫都與地球上現代的 CG風格不同,但希里的畫作中喬修沒有找到一絲、覆古'的感覺。

"你將來真的沒考慮過從事繪畫方面的職業嗎?"

喬修上下檢視著吉安娜 普羅德摩爾的原畫,一些細小的細節還有不少瑕疵,但這些經過喬修的修改還是能 夠接受。

如果希里按照這個進度繼續練習下去,希里一定能成為一位優秀的遊戲原畫師。

"我...不知道。

< MT >

Jochead nannte ihren Namen, eine Figur, die wie ein Faden durch das gesamte "Welt von Warcraft"-Epos zieht und auch die Hauptdarstellerin des Spiels "Hearthstone" ist. Ebenso ist sie die Vertreterin des Berufs "Magier". "Hast du mit ihr irgendeine Beziehung...?" Hilara schwieg für einen Moment, aber sie konnte sich nicht zurückhalten und die Frage nach Jocheads Antwort. "Beziehung?" Jochead schwieg zunächst überrascht. Jaina war nur ein fiktiver Charakter im Spiel, und Jochead hatte mit ihr keine Beziehung. Als "Welt von Warcraft"-Spielzeugspielespieler fühlte Jochead jedoch, dass solch eine Antwort nicht ganz angemessen war. "Ich denke, es ist wahrscheinlich, dass wir ehemalige Verbündete waren. "In der Vergangenheit spielte Jochead "Welt von Warcraft"-Spielzeugspielespiele für die Allianz-Alli

Figure 8: Case 2: Failed evaluation on **ARGS**, zh-de, metricX-qe: 8.47, produced incomplete translation and repeated words and sentences

< SRC >

```
今天沒被要求伺候,少了討好皇女的機會,但從情感上來 說,他就算身份低微,也未嘗沒幻想過一個貌美有才的妻主,而不是一個
猥瑣的草包。內心應該慶幸才對...
少年低聲應是,隨即退了出去。但不知為什麼,退到外間停下之後,他又鬼使神差地看了眼屏風。
那後面映出一個曼妙的女子身影,隨意慵懒的穿衣動作,竟讓人有些……面紅心跳。
少年忽然想,其實三皇女也不錯,雖然人很猥瑣,還是個草包,但確實長得漂亮,而且身份高貴啊 ……
以後,還是盡力取悅她吧。他默默下了決心,臉色又紅了一些,低下頭,恭敬地垂首站立。
第181章 女尊第一花魁(2)
她把原主的記憶梳理了一遍。在這個女尊男卑的世界,原主的身份也很尊貴,是王朝的三皇女。就是人品差了點,沒用了點,急色
了點,其他還是挺好的 .....嘛。
二皇女早夭,女皇陛下只有大皇女和三皇女兩個女兒,對大皇女管教比較嚴格,對小女兒就是真的溺愛了。這也是三皇女最大的靠
山。要不是女皇寵原主,就憑大皇女的手段,原主早不知道死多少次了。
而原主的心願,也和大皇女有關。
原主是放蕩不羈的性格,愛好各種美人,從沒想過爭奪王位,只想等大姐繼位之後,當個閒散的親王。大皇女卻不放過她。
這位好姐姐知道妹妹愛美男,特意派了個美少年引誘她,三皇女被對方的氣質與美貌迷得神魂顛倒,最後被哄得一杯毒酒下肚,才
得知真相。
美少年是她姐姐的人,並不喜歡她,不過是逢場作戲,虛與委蛇而已。她敬重的姐姐,更是要她的命。所以她不甘心。
她的心願就是報覆這對男女。她要讓為了繼位殺害自己的大姐,這輩子都得不到皇位,讓那個欺騙自己感情的少年,這輩子都得不
喻楚想,這可咋整。皇位還算簡單,但愛情?她怎麼確保那人得不到愛情?劇情都不 說了嗎,人家和大皇女兩情相悅啊!
```

< MT >

She flipped through the memories of the original owner, crying and laughing in dismay. "I'm glad he's still in his first affair. He's definitely much better off than he was in his past life."

She organised the memories of the original owner, recalling his life in a world of female dominance and male servitude. The original owner had been the king's third princess, second in line to the throne. While she had been spoiled by her mother, the empress, her older sister, the second princess, had been given strict treatment.

In her memories, the third princess was just as flawed as her sister, lacking any kind of sense of loyalty or duty, prone to whims and impulsiveness.

Figure 9: Case 3: Failed evaluation on **LLaMA3.1-8B**, zh-en, metricX-qe: 9.85, did not start the translation from the beginning and the meaning is incomplete.