Hybrid Synthesis Tag and Two-Stage Training for LLM-Enhanced Tag-Aware Machine Translation

Anonymous ACL submission

Abstract

Internet texts often contain rich format tags that carry structural, semantic, and functional meaning. However, current large language model (LLM)-based translation systems struggle to balance translation fluency with tag structure preservation when processing tagged text. To address this, we propose a tag-aware machine translation optimization framework leveraging LLM, which aims to simultaneously enhance translation quality and preserve tag structures. First, we design a hybrid synthesis tag strategy to generate structurally complex and high-quality tagged training data. We then introduce a two-stage tag-aware training framework: in the first stage, supervised fine-tuning is conducted through multi-task learning to enable the model to deeply understand tag semantics and their scope; in the second stage, a multi-reward mechanism is introduced for reinforcement learning, using fine-grained reward functions to optimize translation adequacy and tag preservation performance. For comprehensive evaluation, we construct a multilingual tag translation test set with complex tag structures (to be open-sourced) and propose a comprehensive evaluation metric. Experimental results demonstrate that our method significantly outperforms existing approaches in both translation quality and tag structure consistency.

1 Introduction

In the context of deep integration between globalization and the internet, cross-lingual information exchange is becoming increasingly frequent. Web pages, documents, and various other forms of online content widely adopt structured tags such as HTML. These tags not only control the presentation format of text but also carry rich semantic connotations and functional value. During machine translation (Vaswani et al., 2017; Sutskever et al., 2014), if such tags are not properly handled, it can easily lead to formatting chaos and may further cause semantic distortion and functional failure, thereby compromising the accuracy of information delivery and the ultimate user experience.

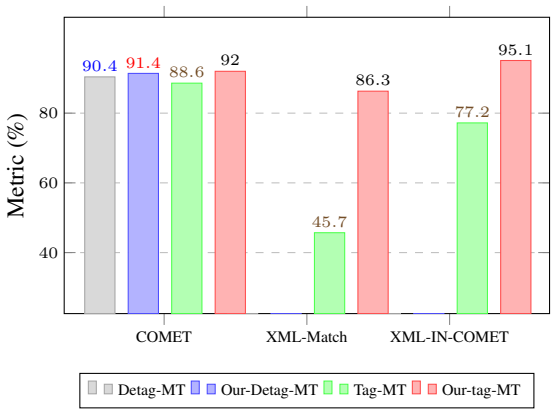

Figure 1: Our LLM-Enhanced tag-aware translation vs. plain SFT: A Comparison of Detag and Tag Translation Modes in English-Chinese machine Translation.

Although large language models (LLMs) (Touvron et al., 2023; Bai et al., 2023; DeepSeek-AI, 2024) make significant progress in translating untagged text, they still face considerable challenges when handling tagged text. As shown in Figure 1, after supervised fine-tuning (SFT) (Dong et al., 2024) on untagged data (Plain SFT), LLMs not only exhibit low consistency in tag structure but also a decline in translation quality when directly translating tagged text. Therefore, tag-aware translation for LLMs are highly significant to explore.

Before the rise of LLMs, tag-aware translation primarily relied on two approaches: one is the detag-and-project method (Joanis et al., 2013; Müller, 2017; Zenkel et al., 2021), which involves removing tags from the source text for translation and then mapping source-language tags to the translated text based on alignment. This method separates the translation process from tag handling, making it prone to error propagation and leading to semantic misalignment between tags and translations or structural nesting issues. The other is

the masked tag training method (Hanneman and Dinu, 2020; Elshin et al., 2024), which involves normalizing tags to incorporate them into the translation process. However, due to the sparsity of tags in training data, the effectiveness of normalization remains limited.

With the development of LLMs, related studies (Dabre et al., 2023) have begun to introduce tagged bilingual sentence pairs as few-shot example to aid translation. Although such methods are relatively easy to implement, their effectiveness still falls short of training models directly on tagged data. Other studies (Dabre et al., 2024; Ryu et al., 2022) have leveraged phrase alignment techniques or utilized LLMs to automatically add tags to untagged bilingual texts, thereby constructing tagged bilingual training data to enhance translation model capabilities. However, the lack of diversity in synthetic data makes it challenging for models to effectively handle complex and varied tag structures. Overall, existing methods remain crude in data synthesis and lack systematic modeling and optimization for tag-aware training.

To address this, we propose a LLM-based tag-aware translation method. This method integrates a hybrid tag synthesis strategy with a two-stage training pipline, aiming to enable LLM not only to perform text translation but also to deeply understand the functional intent behind tags, thereby achieving accurate and flexible semantic and functional conversion in the target language.

2 Method

As shown in Figure 2, our proposed method consists of two parts: hybrid synthetic tag and two-stage tag-aware training.

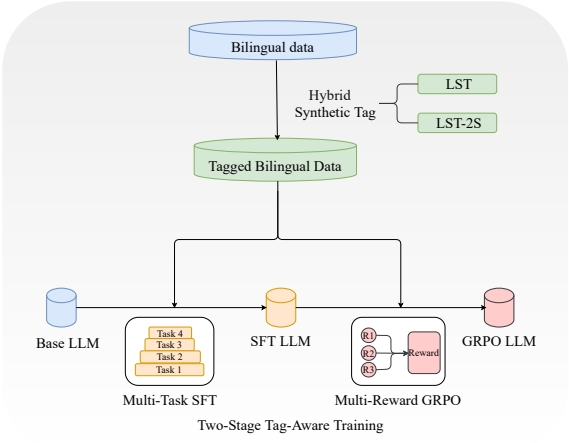

Figure 2: Our Proposed Tag-Aware Translation Method.

2.1 Hybrid Synthetic Tag

Tagged bilingual data serves as the foundation for tag-aware training, but such data is often scarce and requires synthesis. A common practice involves leveraging abundant bilingual data and injecting tags into it. Currently, there are two mainstream methods for synthesizing tags:

Phrase Alignment-Based Synthesis Tag (AST) (Ryu et al., 2022): First, bilingual word alignment is obtained using alignment tools, then expanded into phrase alignment. Finally, tags are sampled from a predefined set and wrapped around randomly selected aligned phrase pairs in the source and target languages.

LLM-based Synthesis Tag (LST) (Dabre et al., 2024): Using a few tagged examples as prompts, LLM can be guided to directly generate tagged bilingual sentence pairs from untagged ones.

However, both existing methods have limitations: AST tends to produce unnatural phrase structures due to random tag insertion, while LST, although capable of selecting more natural insertion positions, remains constrained by alignment with the target side, resulting in relatively rigid generated data that affects model generalization. To address these issues, we propose the following enhanced methods based on LLM:

Two-Stage LLM-based Synthesis Tag (LST-2S): LST-2S method comprises two stages. In the first stage, guided by a small set of examples, LLM is prompted to sample tags from a predefined set and insert them randomly into the source text, without imposing alignment constraints with the target, thereby enriching structural diversity. In the second stage, the original bilingual text is used as contextual guidance for LLM to translate the tagged source text into corresponding tagged target text. This allows flexible adaptation of sentence structures, improving expressiveness and naturalness. Figure 3 shows examples of synthesis tag using different methods on English-Chinese (en2zh).

Hybrid LLM-based Synthesis Tag (Hy-LST): To address potential translation quality degradation in LST-2S, we further propose a hybrid strategy that integrates LST with LST-2S, aiming to balance translation quality with tag structural diversity.

2.2 Two-Stage Tag-Aware Training

Tag-aware training enhances the translation of tagged texts by fine-tuning a general LLM on tagged bilingual data. The process begins with

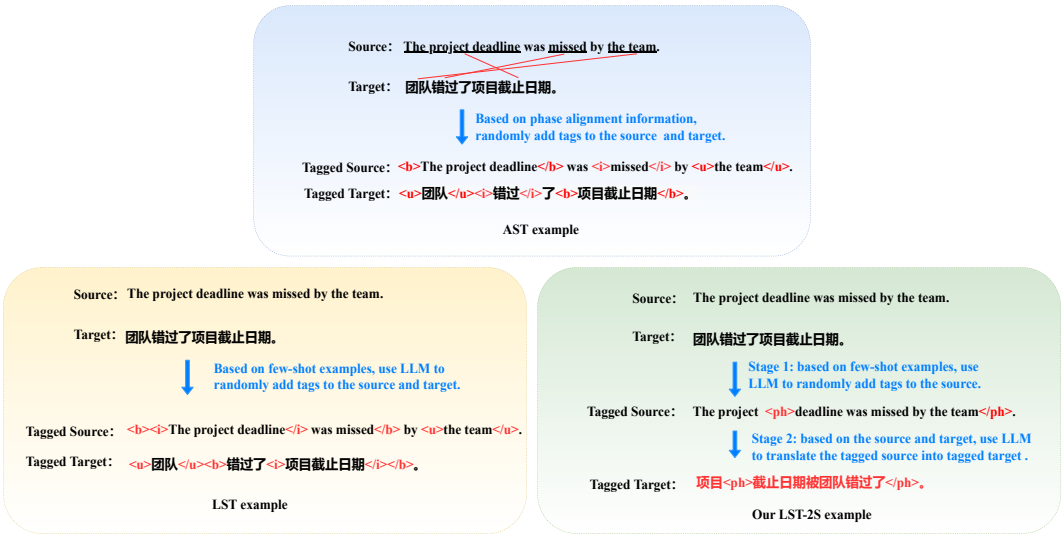

Figure 3: Our method (LST-2S) vs. existing methods (AST and LST) for synthesizing en2zh tag examples.

SFT (Dong et al., 2024), which enables the model to properly process tagged text. For better performance, we propose a two-stage training strategy, consisting of multi-task SFT followed by multi-reward Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

2.2.1 Multi-Task SFT

As shown in Figure 4, during the multi-task SFT stage, following the design concept of decoupling and reconstruction, we decompose the tag-aware translation task into four progressively advanced training tasks, aiming to enable the model to fully master the ability to translate tagged text. The specific task definitions are as follows:

TASK1 Prompt:
Translate the following untagged sentence from {source_lang} to {target_lang}.
untagged {source_lang}: {source_sentence}
untagged {target_sentence}:

TASK2 Prompt:
Translate the following tagged sentence from {source_lang} to {target_lang}.
tagged {source_lang}: {tagged_source_sentence}
tagged {target_sentence}:

TASK3 Prompt:
Reference the untagged {source_lang} sentence, translate the following tagged sentence from {source_lang} to {target_lang}.
untagged {source_lang}: {untagged_source_sentence}
tagged {source_lang}: {tagged_source_sentence}
tagged {target_sentence}:

TASK4 Prompt:
Reference the untagged {source_lang} and {target_lang} sentence pair, translate the following tagged sentence from {source_lang} to {target_lang}.
untagged {source_lang}: {untagged_source_sentence}
untagged {target_lang}: {untagged_target_sentence}
tagged {source_lang}: {tagged_source_sentence}
tagged {target_sentence}:

Figure 4: Prompt Design for Multi-task SFT.

Untagged Translation (Task1): Training is conducted based on untagged bilingual parallel data, aiming to establish fluent semantic mapping relationships and build a foundational translation framework for subsequent tasks.

Tagged Translation (Task2): Training utilizes tagged bilingual data, introducing tags in the form of special tokens. This enables the model to learn to recognize the opening and closing structures of tags, construct a tag vocabulary, and thus grasp the basic usage patterns of tags in the text.

Tagged Translation with Untagged Source as Context (Task3): Building on the tagged bilingual data, untagged source language text is additionally introduced as context to help the model understand the semantic function and scope of tags.

Tagged Translation with Untagged Source and Target as Context (Task4): Continuing with the use of tagged bilingual data, both untagged source language text and target language text are introduced as context. This guides the model to accurately insert tags in the translation and flexibly adjust sentence structures based on the context.

2.2.2 Multi-Reward GRPO

After completing SFT, to further calibrate the model's tag-aware translation performance, we design three reward functions based on the GRPO reinforcement learning framework (Shao et al., 2024), each assigned an identical weight coefficient, to systematically improve the translation quality and tag structure consistency of tagged texts. The definitions of each reward function are as follows:

Translation Quality Reward (R1): As shown in Equation 1, we use COMET (Rei et al., 2020, 2022a) as the reward function to evaluate the overall translation quality after tag removal. Here, src_{detag}, mt_{detag} and ref_{detag} denote the tag-removed versions of the source sentence, the

translated output, and the reference translation, respectively. To mitigate overfitting, we adopt COMET-20 during the GRPO training phase, and switch to COMET-22 for the testing phase.

$$R_1 = comet_{20}(src_{detag}, mt_{detag}, ref_{detag}) \quad (1)$$

Tag Structure Consistency Reward (R2): As shown in Equation 2, we assess the tag structure consistency between the translation and the reference. Here, $tags_{mt}$ denotes the tag set of the translation, while $tags_{ref}$ represents the tag set of the reference. The reward R_2 takes a value of 1 only if the two sets are identical; otherwise, it is 0.

$$R_2 = 0 \ \ if \ \ tags_{mt} \neq tags_{ref} \ \ else \ \ 1 \quad (2)$$

Tag-Scoped Translation Quality Reward (R3): As shown in Formula 3, we evaluate the translation quality of content within each tag pair using our XML-IN-COMET. Here, N represents the total number of tag pairs; src_i, mt_i, and ref_i respectively denote the source, translation and reference within i-th tag pair. The reward R_3 is the average of the translation quality scores across all tag pairs.

$$R_3 = mean(\{comet_{20}(src_i, mt_i, ref_i)\}_{i=1}^{N}) \quad (3)$$

3 Experiment

3.1 Dataset

3.1.1 Open-source Dataset

The dataset serves as the foundation for the training and evaluation of tag-aware translation tasks. Previously, Hashimoto et al. (Hashimoto et al., 2019) constructed a high-quality multilingual dataset[1] for structured document translation, which has now become an open-source benchmark in this field. Based on this, we select five translation directions for our experiments: English to Chinese, Japanese, German, French, and Russian.

Each translation direction contains approximately 100,000 training samples, along with 2,000 development set samples and 2,000 test set samples. Since the references for the test set are not publicly available, we use the development set as the test set for evaluation. To avoid introducing model bias due to duplication between training data and test data, we deduplicate the source texts in the training sets of all translation directions against the source texts in each test set. Subsequently, we randomly select 2,000 samples from the training set of each direction as the development set. Detailed data statistics are shown in Table 1.

Dataset	**en2zh**	**en2ja**	**en2de**	**en2fr**	**en2ru**
Train	88611	90761	91333	91419	88983
Dev	2000	2000	2000	2000	2000
Test	2000	2000	2000	2000	2000

Table 1: Our extended localization-xml-mt dataset.

3.1.2 Extended Testset

Evaluation baselines for open-source datasets show that both XML-ACC and XML-Match metrics exceed 99% on the original test set. On the surface, the tag-aware translation task appears to require no further optimization. However, analysis reveals that only 27% of the original test set contains tags, and 82% of those samples contain only 1–2 tag pairs, indicating relatively low difficulty. To increase the challenge of the test, we employ the LST method to augment tags in the test set: first, we manually construct three examples with added tags for each translation direction; then, multiple LLMs (Qwen-max (Bai et al., 2023), GPT-4o (Hurst et al., 2024), and Claude-4) are called to randomly add tags to the test set based on the provided examples; finally, qualified data are filtered based on XML-MATCH and XML-IN-KIWI metrics (with thresholds set at 100 and 80, respectively), while samples that remain unqualified after multiple attempts retain their original data. After expansion, the proportion of samples containing tags significantly increases from 27% to 97%, and the proportion of samples with only 1–2 tag pairs drops from 82% to 44%, substantially enhancing the complexity and reliability of the test set. To avoid convergence in data distribution with the extended testset, we use DeepSeek-v3 (DeepSeek-AI, 2024) when synthesizing tagged training data with LLMs.

3.2 Evaluation Metrics

Our evaluation of tag-aware translation primarily focuses on three aspects: translation quality, tag structure consistency, and overall comprehensive evaluation. Regarding translation quality, we adopt BLEU (Papineni et al., 2002) and COMET (Rei et al., 2022a) metrics to assess the translation quality of pure text after tags are removed. For tag structure consistency, we use XML-ACC and XML-Match (Hashimoto et al., 2019) to measure the correctness of the tags themselves in the translation, as well as their correspondence with the

[1] https://github.com/salesforce/localization-xml-mt

tags in the source text. In terms of comprehensive evaluation, early methods employed XML-BLEU (Hashimoto et al., 2019), the basic idea of which is to split the translation into multiple lines according to tags and compute BLEU line by line.

However, when a translation contains multiple tags, this method may lead to alignment deviations due to tag misplacement, affecting evaluation accuracy. Therefore, we propose a simplified splitting strategy: only extracting the text content between each pair of tags to avoid alignment issues, and then using BLEU to evaluate the quality of these segments. We name this metric XML-IN-BLEU. Furthermore, by replacing BLEU with COMET or KIWI (Rei et al., 2022b), it can be extended to XML-IN-COMET and XML-IN-KIWI to meet more flexible evaluation needs. During the testing phase, COMET calculations use COMET22 to avoid overlap with the GRPO training phase.

3.3 Baselines

We choose Qwen2.5-7B-Instruct as the base LLM to carry out experiments. Below are the comparison baselines we build based on existing methods:

Prompt Baseline: We take tagged text as input and perform translation by prompting the base LLM.

Few-shot Prompt Baseline: Based on LABSE semantic similarity (Feng et al., 2022), we select the three most similar source-target example sentences from the training set for each test sentence to construct few-shot translation examples, guiding the base LLM to translate the tagged text.

Detag-and-project Baseline: We remove the tags from the source language sentence and translate it using the Plain SFT LLM, which is obtained by supervised fine-tuning of the base LLM on tag-free corpora. Then, we automatically extract bilingual word-level alignment relations using the awesome-align tool (Dou and Neubig, 2021), implemented based on multilingual BERT (Wang et al., 2019). Before processing, the text must be tokenized: English, French, German, and Russian are tokenized using the Moses tokenizer[2], Chinese using jieba[3], and Japanese using Mecab[4]. Finally, based on the word alignment results, the tags are inserted into corresponding positions in the translation using Min-Max Tag Pair Projection(Zenkel et al., 2021).

Masked SFT Baseline: During the model training phase, we replace the actual tags in the original tagged data with numbered Mask tokens for training; during the inference phase, we first replace the tags in the input text with the corresponding Mask tokens, and after the model generates the translation results, we restore the Mask tokens in the translation to the actual tags.

AST SFT Baseline: When synthesizing tagged data using the AST method and then conducting SFT training, the key is to extract matching phrase pairs for each parallel sentence pair to build aligned tagged data. The specific steps are as follows: first, use the awesome-align tool to perform word-level alignment on the bilingual data. Then, based on the word alignment results, apply a phrase extraction algorithm (Och et al., 1999) to obtain all phrase pairs that comply with word alignment constraints. This algorithm determines phrase-level correspondences by enumerating all possible phrase combinations consistent with the word alignment; a concrete implementation can refer to the relevant algorithm in the NLTK library (Bird and Loper, 2004). Finally, for each sentence pair and its corresponding set of phrase alignments (where the number of alignments typically far exceeds the number of words in the sentences), we randomly select a subset of aligned phrase pairs and wrap them with a tag to synthesize the required tagged training data.

LST SFT Baseline: We use the LST method to synthesize tagged data and then perform supervised fine-tuning. Specifically, we select the Deepseek-V3 model and manually create three few-shot examples per translation direction. These examples guide the model in generating source and target sentences with tags from sentences without tags.

3.4 Our Baselines

Besides existing methods, we also construct the following two new LLM-based baselines:

LST-2S SFT Baseline: LST-2S SFT baseline and LST SFT baseline both utilize synthetic tag data generated from Deepseek-V3, but they differ in their tag generation processes. LST-2S adopts a two-stage synthetic tagging approach: first, we manually create three few-shot examples per translation direction to instruct Deepseek-V3 in tagging the source text; second, we prompt Deepseek-V3 to translate the tagged source text by referencing the untagged source and target sentences.

Hy-LST SFT Baseline: Hy-LST SFT Baseline improves model training effectiveness by integrating the advantages of both LST and LST-2S tag synthesis methods. In the specific implementation,

[2] https://github.com/moses-smt/mosesdecoder
[3] https://github.com/fxsjy/jieba
[4] https://github.com/SamuraiT/mecab-python3

en2zh	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Prompt	32.99	84.93	94.25	38.15	33.79	76.83	60.16
Few-shot Prompt	34.07	85.55	97.05	60.15	49.59	84.32	68.46
Detag-and-project	55.49	90.36	95.25	73.65	56.53	90.03	76.89
Masked SFT	55.84	90.99	99.85	79.30	77.01	94.02	82.84
AST SFT	55.70	90.75	99.15	80.70	75.44	93.94	82.61
LST SFT	57.16	91.33	99.95	85.10	78.22	94.48	84.37
LST-2S SFT	56.06	91.13	99.90	82.90	77.66	94.14	83.63
Hy-LST SFT	57.40	91.41	99.95	85.21	78.70	94.64	84.55
Multi-Task Hy-LST SFT	59.66	91.76	99.95	85.00	79.93	94.94	85.21
+Multi-Reward GRPO	**60.22**	**91.98**	**100.00**	**86.25**	**80.16**	**95.10**	**85.62**

Table 2: Evaluation results of different methods on the the en2zh ocalization-xml-mt extended testset.

untagged training data is equally divided into two parts: one part utilizes the LST method, while the other employs the LST-2S method. Finally, the two parts of data are merged to train the base LLM.

3.5 Training Details

We report the implementation details and hyperparameter settings during the training phase.

3.5.1 SFT

We utilize Qwen2.5-7B-Instruct as the base model and implement SFT and multi-task SFT training based on the LlamaFactory framework (Zheng et al., 2024). The specific configuration is as follows: we adopt LoRA (Gao et al., 2024) for fine-tuning, with the rank set to 8, the scaling parameter α set to 16, and all LoRA target modules are trained. The LoRA dropout (Srivastava et al., 2014) is set to 0.0. During training, the warmup (Fradkin et al., 2010) ratio is set to 0.1, the learning rate scheduler adopts a cosine annealing strategy (Liu, 2022), and the learning rate is set to 1e-4. Gradient accumulation is used to achieve an effective batch size of 32. The model is trained for up to 3 epochs, with validation evaluation conducted after each epoch, and the checkpoint with the lowest validation loss is selected as the final optimal model. All experiments are performed on 8 GPUs.

3.5.2 GRPO

We conduct Mutil-Reward GRPO training on the model after SFT based on the Open-R1 framework (Hugging Face, 2025). To improve training efficiency, we choose Task2 as the training objective. This is mainly because Task2 has a faster translation speed and shows minimal performance gaps compared to other tasks after multi-task Hy-LST SFT. The data amount used in this phase is approximately one-tenth of that used in the SFT stage. The training is conducted in parallel across 8 GPUs: one GPU is responsible for running the vLLM (Kwon et al., 2023) inference engine to generate 7 candidate translations for each source text at a temperature of 1.0, while the remaining 7 GPUs handle the training tasks, each with a batch size of 7 per card and an equivalent batch expansion achieved through 4-step gradient accumulation. To enhance memory and computational efficiency, we employ the DeepSpeed (Rasley et al., 2020) ZeRO3-offload optimization strategy along with bfloat16 (Kalamkar et al., 2019) mixed-precision training. The main training parameters are as follows: the learning rate is set to a relatively low value of 1e-6, the scheduler adopts a cosine annealing strategy with a warm-up step count of 5% of the total training steps, and the maximum sequence length is limited to 4096 tokens. The reward signal used for model updates is obtained by averaging the outputs of three reward functions. We implement an early stopping mechanism based on validation loss: if the loss does not decrease over five consecutive evaluations, the training is terminated. Evaluations and model checkpoint saving are performed every 100 steps.

4 Result and Analysis

4.1 Main Results

Table 2 presents the results of of our tag-aware method compared with existing baselines on en2zh, while the results for the other four translation directions are detailed in Appendix A. Among the existing baseline methods, the LST SFT baseline achieves the best overall performance. In our new baseline approaches, limited by the quality of regenerated translations during tag synthesis, the LST-2S baseline fails to surpass the original LST baseline. However, the Hybrid-LST SFT baseline, obtained by mixing LST-2S with LST, outperforms all existing baselines across nearly all evaluation metrics. Furthermore, based on the aforementioned Hybrid-LST data, our proposed

en2zh	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Plain SFT	51.54	88.57	98.45	45.70	33.66	77.23	65.86
Clean SFT	55.29	90.60	99.90	77.20	76.18	93.34	82.09
Masked SFT	55.84	90.99	99.85	79.30	77.01	94.02	82.84
AST SFT	55.70	90.75	99.15	80.70	75.44	93.94	82.61
LST SFT	57.16	91.33	99.95	85.10	78.22	94.48	84.37
LST-2S SFT	56.06	91.13	99.90	82.90	77.66	94.14	83.63
LST+LST-2S SFT	**57.40**	91.41	99.95	**85.21**	**78.70**	94.64	**84.55**
AST+LST-2S SFT	56.45	91.15	99.90	82.60	77.73	94.20	83.67
AST+LST+LST-2S SFT	57.30	**91.42**	**100.00**	84.40	78.56	**94.72**	84.40

Table 3: Results for the en2zh localization-xml-mt extended development set.

two-stage tag-aware training method (Multi-Task SFT + Multi-Reward GRPO) achieves significant performance improvements at each stage.

Translation Quality: BLEU and COMET are used to evaluate translation quality of untagged texts. Without extra training, Prompt and Few-shot Prompt baselines perform far worse than other SFT methods. The Detag-and-project baseline uses a base SFT model trained on untagged data, representing the basic SFT translation ability, with BLEU and COMET scores of 55.49 and 90.36. Applying Masked SFT and AST SFT improves translation quality. Using LST SFT, LST-2S SFT, and Hy-LST SFT brings even greater improvement. Among all baselines, Hy-LST SFT achieves the best results, with BLEU and COMET reaching 57.40 and 91.41. Adding two-stage tag-aware training further boosts performance, raising BLEU and COMET to 60.22 and 91.98.

Tag Structure Consistency: XML-ACC measures the accuracy of tag structures in translations. Most baselines score above 99 here, showing that with simple fine-tuning, LLMs rarely break tag structures when translating tagged text. XML-Match evaluates how consistent tag structures are between translations and source texts. The prompt-based baseline performs worst in tag structure consistency, with XML-Match around 38.15. The few-shot prompt baseline improves significantly but remains relatively low at 60.15. This indicates the base LLM still struggle to keep tag structures consistent between translations and sources. Among existing baselines, this metric can be improved to over 85.21. After two-stage tag-aware training, XML-Match is further raised to 86.25.

Overall comprehensive evaluation: XML-IN-BLEU and XML-IN-COMET assess translation quality within each tag pair, evaluating both the accuracy of tag scope and the quality of content translation. Errors in either dimension result in lower scores. Among all baselines, Hy-LST SFT achieves the highest performance, with scores of 78.70 on XML-IN-BLEU and 94.64 on XML-IN-COMET. Following two-stage tag-aware training, these scores improve to 80.16 and 95.10, respectively. The evaluation trends observed are consistent with those from other metric categories, providing further validation of the assessment results.

4.2 Comparison with Tag Synthetic Methods

We compare different tag synthesis methods in SFT. First, two baselines are introduced: Plain SFT uses bilingual data with tags removed, while Clean SFT retains original tag structures. We also explore three hybrid tag synthesis approaches in Hybrid-LST SFT: LST + LST-2S, AST + LST-2S, and AST + LST + LST-2S. When synthesizing tag data using mixed methods, we keep the total amount of training data unchanged and only distribute the data evenly across different synthesis approaches.

Results show that Plain SFT performs worst in tag translation, with an average score of 65.86, due to no tag information during training. Clean SFT uses manually annotated tags, raising the average score to 82.09. However, since tags occupy a limited proportion in the data, its performance does not surpass other SFT methods that employ synthetic tags. Msked SFT shows that normalizing tags improves overall performance.

Among single-method baselines, LST SFT and LST-2S SFT outperform AST SFT in tag translation, with LST SFT scoring highest at 84.37. Among Hybrid-LST SFT methods, mixing LST + LST-2S works best, achieving 84.55 and surpassing single-method baselines. The experimental results indicate that introducing tag data synthesized by the AST method does not provide significant performance gains for Hybrid-LST SFT.

4.3 Comparison with SFT Training Tasks

We employ the Hybrid-LST data and compare the effects of multi-task joint training versus indepen-

en2zh	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Task1 SFT	51.54	88.57	98.45	45.70	33.66	77.23	65.86
Task2 SFT	57.40	91.41	99.95	85.21	78.70	94.64	84.55
Task3 SFT	58.02	91.58	100.00	85.10	79.19	94.70	84.77
Task1+Task4 SFT	59.39	91.79	99.90	85.40	80.13	94.78	85.23
Multi-Task SFT (Task1 MT)	57.63	91.40	-	-	-	-	-
Multi-Task SFT (Task2 MT)	59.66	91.76	99.95	85.00	79.93	94.94	85.21
Multi-Task SFT (Task3 MT)	60.16	**91.92**	**100.00**	85.65	79.95	94.96	85.44
Multi-Task SFT (Task1+Task4 MT)	**60.18**	91.90	99.95	**85.85**	**80.22**	**95.04**	**85.52**

Table 4: Results for the en2zh localization-xml-mt extended development set.

en2zh	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Multi-Task Hy-LST SFT	59.66	91.76	99.95	85.00	79.93	94.94	85.21
+R1 GRPO	**60.32**	**92.10**	100.00	85.10	80.02	95.01	85.43
+R2 GRPO	59.72	91.81	**100.00**	**86.34**	80.03	95.03	85.49
+R3 GRPO	60.12	91.91	99.95	85.66	**80.19**	**95.14**	85.50
+Multi-Reward GRPO	60.22	91.98	**100.00**	86.25	80.16	95.10	**85.62**

Table 5: Results for the en2zh localization-xml-mt extended development set.

dent training for each task during the SFT stage. For the independent training of Task4, we additionally incorporate data from Task1, as Task4 involves a two-stage translation whose output depends on the translation result of Task1 as input.

The experimental results show that Task1, trained solely on untagged bilingual data, performs the worst. Task2, trained directly on tagged bilingual data, achieves a significant performance improvement, with the average score increasing from 65.86 to 84.55. Task3 further introduces untagged source text as context, leading to even better results, with the average score rising further to 84.77. Meanwhile, Task4, which incorporates both untagged source text and untagged translation as context, demonstrates superior performance, with the average score improving to 85.23.

Multi-task SFT enables mutual enhancement among tasks, leading to further performance gains. Additionally, the model obtained through multi-task SFT supports multiple translation modes. Among the three modes available for translating tagged source text, we evaluate them accordingly: compared to their independently trained counterparts, all three inference modes under multi-task SFT show improved performance, with the gaps between them significantly narrowing. Among these, the translation mode for Task2 achieves an average score of 85.21, slightly lower than the other twos.

4.4 Comparison with GRPO Rewards

Based on the multi-task Hy-LST SFT LLM, we further compare the effects of jointly using versus separately using three reward functions during the GRPO training phase. Experimental results show that R1 function primarily improves translation quality, with BLEU and COMET scores reaching 60.32 and 92.10 respectively; R2 function more significantly enhances tag structure consistency, increasing the XML-MATCH score to 86.34; while R3 function achieves better performance on comprehensive evaluation metrics, with XML-IN-BLEU and XML-IN-COMET reaching 80.19 and 95.14 respectively. Overall, using all three reward functions together yields the best overall effect, with the most notable improvement in the average score, rising from 85.21 to 85.62.

5 Conclusion

This paper tackles the challenge of balancing translation fluency and tag structure consistency when LLMs translate tagged texts. It proposes a comprehensive tag-aware machine translation optimization method. By employing innovative data construction and a systematic training framework, the approach significantly improves translation in complex tag scenarios. First, Hybrid-LST is designed to address the scarcity of tagged bilingual data. Second, a two-stage tag-aware training enhances the model's tag comprehension and conversion capabilities. For evaluation, a multilingual test set with complex tag structures and comprehensive metrics is constructed. Experiments show our method outperforms existing baselines in both translation quality and tag structure consistency, validating the hybrid data strategy and two-stage training. This work provides new insights for handling structured text in machine translation. Future research may extend the framework to other tag-based languages (e.g., LaTeX and Markdown).

Limitations

Although the method proposed in this paper achieves significant progress, it still has some limitations. First, our method heavily relies on the data synthesis and instruction-following capabilities of large language models, and its performance ceiling may be constrained by the performance and biases of the underlying large language model itself. Second, the experiments primarily focus on format tags such as HTML tags in web pages; for tag types involving more complex logic or dynamic functions, the generalization ability of the method requires further validation. Finally, although the constructed evaluation dataset and improved evaluation metrics are highly targeted, the tag scenarios they cover remain limited. Future work should extend to more complex real-world application scenarios for more comprehensive validation.

References

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, and 29 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Raj Dabre, Bianka Buschbeck, Miriam Exel, and Hideki Tanaka. 2023. A study on the effectiveness of large language models for translation with markup. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 148–159.

Raj Dabre, Haiyue Song, Miriam Exel, Bianka Buschbeck, Johannes Eschbach-Dymanus, and Hideki Tanaka. 2024. How effective is synthetic data and instruction fine-tuning for translation with markup using llms? In *Proceedings of the 16th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 73–87.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Denis Elshin, Nikolay Karpachev, Boris Gruzdev, Ilya Golovanov, Georgy Ivanov, Alexander Antonov, Nickolay Skachkov, Ekaterina Latypova, Vladimir Layner, Ekaterina Enikeeva, and 1 others. 2024. From general llm to translation: How we dramatically improve translation quality using human evaluation data for llm finetuning. In *Proceedings of the Ninth Conference on Machine Translation*, pages 247–252.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Andrea J Fradkin, Tsharni R Zazryn, and James M Smoliga. 2010. Effects of warming-up on physical performance: a systematic review with meta-analysis. *The Journal of Strength & Conditioning Research*, 24(1):140–148.

Dehong Gao, Yufei Ma, Sen Liu, Mengfei Song, Linbo Jin, Wen Jiang, Xin Wang, Wei Ning, Shanqing Yu, Qi Xuan, and 1 others. 2024. Fashiongpt: Llm instruction fine-tuning with multiple lora-adapter fusion. *Knowledge-Based Systems*, 299:112043.

Greg Hanneman and Georgiana Dinu. 2020. How should markup tags be translated? In *Proceedings of the Fifth Conference on Machine Translation*, pages 1160–1173.

Kazuma Hashimoto, Raffaella Buschiazzo, James Bradbury, Teresa Marshall, Richard Socher, and Caiming Xiong. 2019. A high-quality multilingual dataset for structured documentation translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 116–127, Florence, Italy. Association for Computational Linguistics.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Eric Joanis, Darlene Stewart, Samuel Larkin, and Roland Kuhn. 2013. Transferring markup tags in statistical machine translation: A two-stream approach. In *Proceedings of the 2nd Workshop on Post-editing Technology and Practice*.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, and 1 others. 2019. A

study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Zhao Liu. 2022. Super convergence cosine annealing with warm-up learning rate. In *CAIBDA 2022; 2nd International Conference on Artificial Intelligence, Big Data and Algorithms*, pages 1–7. VDE.

Mathias Müller. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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*, pages 3505–3506.

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

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

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

Yonghyun Ryu, Yoonjung Choi, and Sangha Kim. 2022. Data augmentation for inline tag-aware neural machine translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 886–894.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *NIPS*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NIPS*.

Zihan Wang, Stephen Mayhew, Dan Roth, and 1 others. 2019. Cross-lingual ability of multilingual bert: An empirical study. *arXiv preprint arXiv:1912.07840*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2021. Automatic bilingual markup transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533.

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)*, Bangkok, Thailand. Association for Computational Linguistics.

A Main Results for the other four translation directions

As shown in Table 6,7,8,9, we compare the performance of the tag-aware method against existing baselines across four translation directions: en2ja, en2de, en2fr, and en2ru. The results show that our method achieves consistent improvements across all translation directions.

en2ja	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Prompt	30.97	82.04	93.30	30.90	22.62	73.86	55.62
Few-shot Prompt	37.20	84.82	98.00	64.15	39.14	84.28	67.93
Detag-and-project	60.18	92.38	93.95	65.60	41.02	88.21	73.56
Masked SFT	60.22	92.79	99.80	73.85	68.81	93.84	81.55
AST SFT	59.97	92.27	99.15	73.45	67.82	93.67	81.06
LST SFT	62.24	93.06	**100.00**	81.45	71.58	94.79	83.85
LST-2S SFT	61.46	92.97	99.90	77.50	70.66	94.21	82.78
Hy-LST SFT	62.24	93.20	99.95	81.45	72.24	94.83	83.99
Multi-Task Hy-LST SFT	65.23	93.61	**100.00**	83.50	75.01	95.20	85.43
+Multi-Reward GRPO	**65.62**	**93.76**	99.95	**83.75**	**75.88**	**95.47**	**85.74**

Table 6: Evaluation results of different methods on the the en2ja ocalization-xml-mt extended testset.

en2de	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Prompt	25.84	76.35	92.90	43.80	28.74	74.91	57.09
Few-shot Prompt	30.32	81.18	94.60	75.20	42.04	82.75	67.68
Detag-and-project	51.14	89.53	97.40	81.70	50.50	89.34	76.60
Masked SFT	48.46	88.73	99.40	87.00	63.65	90.75	79.67
AST SFT	48.45	88.62	99.25	86.85	62.92	90.41	79.42
LST SFT	52.73	89.53	**100.00**	91.55	67.87	92.74	82.40
LST-2S SFT	51.28	89.68	99.90	88.95	66.04	92.45	81.38
Hy-LST SFT	52.91	89.65	**100.00**	92.00	68.17	92.74	82.58
Multi-Task Hy-LST SFT	56.72	90.34	**100.00**	91.35	71.15	93.41	83.83
+Multi-Reward GRPO	**57.35**	**90.64**	**100.00**	**92.25**	**71.98**	**93.68**	**84.32**

Table 7: Evaluation results of different methods on the the en2de ocalization-xml-mt extended testset.

en2fr	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Prompt	36.89	80.81	92.05	46.05	33.32	74.06	60.53
Few-shot Prompt	43.73	81.41	96.40	71.65	47.31	82.15	70.44
Detag-and-project	65.05	89.98	97.90	86.20	61.37	88.84	81.56
Masked SFT	63.84	89.29	99.55	91.50	73.24	91.44	84.81
AST SFT	63.01	88.98	98.45	90.35	72.66	90.98	84.07
LST SFT	65.95	89.79	99.95	96.25	77.00	93.18	87.02
LST-2S SFT	65.73	89.9	**100.00**	94.35	76.89	92.88	86.63
Hy-LST SFT	66.43	90.04	99.95	95.90	77.21	93.29	87.14
Multi-Task Hy-LST SFT	68.87	90.57	**100.00**	96.00	79.25	93.73	88.07
+Multi-Reward GRPO	**69.91**	**90.93**	99.95	**96.52**	**79.99**	**93.92**	**88.54**

Table 8: Evaluation results of different methods on the the en2fr ocalization-xml-mt extended testset.

en2ru	BLEU	COMET	XML-ACC	XML-MATCH	XML-IN-BLEU	XML-IN-COMET	Avg.
Prompt	24.32	80.99	93.30	54.25	23.55	71.66	58.01
Few-shot Prompt	27.58	82.43	96.60	66.05	33.41	77.55	63.94
Detag-and-project	44.45	90.02	96.05	80.10	44.83	86.60	73.68
Masked SFT	42.45	89.57	99.90	86.95	62.03	89.80	78.45
AST SFT	41.72	89.45	99.75	86.55	61.59	89.56	78.10
LST SFT	49.48	90.66	99.95	92.5	67.24	91.89	81.95
LST-2S SFT	46.42	90.54	**100.00**	90.20	65.28	91.31	80.63
Hy-LST SFT	49.82	90.81	99.95	92.55	68.04	91.85	82.17
Multi-Task Hy-LST SFT	52.17	91.24	**100.00**	93.65	69.81	92.42	83.22
+Multi-Reward GRPO	**53.27**	**91.60**	**100.00**	**94.15**	**71.03**	**92.68**	**83.79**

Table 9: Evaluation results of different methods on the the en2ru ocalization-xml-mt extended testset.