DeMPT: Decoding-enhanced Multi-phase Prompt Tuning for Making LLMs Be Better Context-aware Translators

Anonymous ACL submission

Abstract

Generally, the *decoder-only* large language models (LLMs) are adapted to context-aware neural machine translation (NMT) in a concatenating way, where LLMs take the concatenation of the source sentence (i.e., intra-sentence context) and the inter-sentence context as the input, and then to generate the target tokens sequentially. This adaptation strategy, i.e., concatenation mode, considers intra-sentence and inter-sentence contexts with the same priority, despite an apparent difference between the two kinds of contexts. In this paper, we propose an alternative adaptation approach, named **D**ecoding-**e**nhanced **M**ulti-phase **P**rompt **T**uning (DeMPT), to make LLMs discriminately model and utilize the inter- and intra-sentence context and more effectively adapt LLMs to context-aware NMT. First, DeMPT divides the context-aware NMT process into three separate phases. During each phase, different continuous prompts are introduced to make LLMs discriminately model various information. Second, DeMPT employs a heuristic way to further discriminately enhance the utilization of the source-side inter- and intra-sentence information at the final decoding phase. Experiments show that our approach significantly outperforms the concatenation method, and further improves the performance of LLMs in discourse modeling. We will release our code and datasets on GitHub.

1 Introduction

Context-aware neural machine translation (NMT) goes beyond sentence-level NMT by incorporating inter-sentence context at the document level (Zhang et al., 2018; Miculicich et al., 2018; Voita et al., 2018, 2019b,a; Bao et al., 2021; Sun et al., 2022), aiming to address discourse-related challenges such as zero pronoun translation (Wang et al., 2019), lexical translation consistency (Lyu et al., 2021, 2022), and discourse structure (Hu and Wan, 2023). A recent paradigm shift has been witnessed in context-aware NMT with the emergence of the *decoder-only* large language models (LLMs) (BigScience, 2022; Google, 2022; MetaAI, 2023b,a; OpenAI, 2023). These generative language models, trained on extensive public data, have gained significant attention in the natural language processing (NLP) community. In adapting LLMs to context-aware NMT, a common strategy involves concatenating multiple source sentences as a prefix and generating translations token-by-token, relying on the prefix and previously predicted target tokens, as shown in Figure 1 (a). However, a critical observation of this strategy reveals a potential drawback – the equal prioritization of the inter- and intra-sentence contexts during token generation. Importantly, the intra-sentence context inherently contains richer parallel semantic information with the target sentence and should be given a higher priority than the inter-sentence context. Consequently, we propose that separately modeling and utilizing the inter- and intra-sentence contexts should explicitly inform LLMs of the document-level context and the current sentence itself, thus being able to prevent the misallocation of attention weights to source-side tokens (Bao et al., 2021; Li et al., 2023). Inspired by the success of prompt tuning (Li and Liang, 2021; Liu et al., 2022; Tan et al., 2022), our alternative approach, named Decoding-Enhanced Multi-phase Prompt Tuning (DeMPT), aims to enhance LLMs' adaptability to context-aware NMT, as shown in Figure 1 (b).

Specifically, we divide the whole procedure of context-aware NMT into three phases: inter-sentence context encoding, intra-sentence context encoding, and decoding. Following Li and Liang (2021); Liu et al. (2022), we sequentially and differentially adapt LLMs for each phase, utilizing phase-specific trainable prompts. This phased tuning method enables LLMs to independently capture and model both inter- and intra-sentence contexts, facilitating a better understanding of their differ-

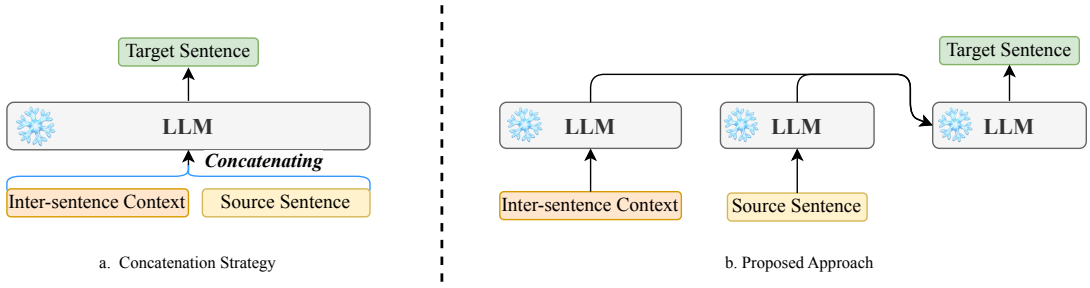

Figure 1: Comparison of different strategies for adapting LLMs to context-aware NMT. The concatenation strategy (*left*) treats inter-sentence and intra-sentence (referred to as the "source sentence" context in the figure) with equal importance. In contrast, our approach (*right*) divides context-aware NMT into three distinct phases, enabling LLMs to selectively model and leverage both inter- and intra-sentence contexts.

ences. Importantly, our approach only divides the original input into three parts without significantly increasing computational load. As a result, there is no substantial decrease in inference speed compared to the concatenating method, as detailed in Section 4.3.

Furthermore, during the decoding phase, we propose a heuristic method to emphasize the difference between inter- and intra-sentence contexts, and avoid *long-distance* issue when utilizing inter-sentence context. Specifically, at each decoding step, we use LLMs to predict the next token three times. The decoding states used for each prediction directly concatenate with the representations of two contexts in a discriminative manner. Finally, we combine three probability distributions to search for the next token as the output from the target vocabulary. This method enables LLMs to learn not only to properly capture inter-sentence context in addressing discourse-related issues but also to recognize a difference between inter- and intra-sentence contexts, allowing for effective utilization of both types of contexts.

In summary, our contributions can be outlined as follows:

- We propose a novel multi-phase prompt tuning approach to divide context-aware NMT into three phases, making LLMs aware of the distinction between inter- and intra-sentence contexts.
- We introduce a enhanced decoding method that discriminately utilize both context types. This allows LLMs not only properly capture inter-sentence context in addressing discourse-related issues, but also be aware of the importance of the intra-sentence context.
- We validate our approach using `llama-2-7b` and `bloomz-7b1-mt` as foundation models, demonstrating its effectiveness across five context-aware translation directions. Extensive analyses further highlight the substantial enhancement in LLMs' ability for context-aware NMT.

2 Methodology

In this section, we describe our decoding-enhanced multi-phase approach for adapting LLMs to context-aware NMT in details. Specifically, we break down the whole procedure of context-aware NMT into three phases (Section 2.1), i.e., inter-sentence context encoding, intra-sentence encoding, and decoding. Additionally, we discriminatively enhance the utilization of inter- and intra-sentence contexts during the decoding phase (Section 2.2). Finally, we describe our phase-aware prompts and training objective in Section 2.3 and Section 2.4, respectively.

For a given ocument pair $(\mathcal{S}, \mathcal{T})$ with K sentences, we will construct K training instances. Each training instance is denoted as a tuple $(\mathcal{C}, S, T)$. Here $S = x|_1^{|S|}$ represents k-th current source sentence with $|S|$ tokens, i.e., intra-sentence context, and $T = y|_1^{|T|}$ is the k-th target sentence with $|T|$ tokens. $\mathcal{C}$ denotes the z previous sentences of S, i.e., the inter-sentence context of S. We denote the hidden size of the LLM as d, and L as the number of transformer layers within it.

2.1 Multi-phase Encoding and Decoding

We implement our approach based on deep prompt tuning (Li and Liang, 2021; Liu et al., 2022). Next, we use training instance $(\mathcal{C}, S, T)$ as an example to describe the multi-phase approach. Figure 2 illustrates the procedure of multi-phase prompt tuning.

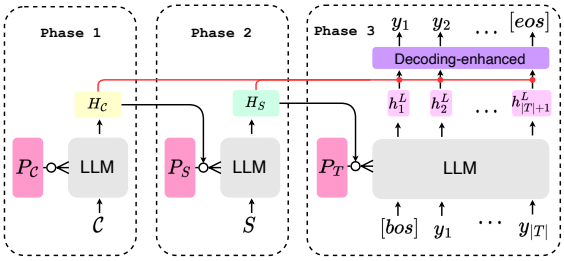

Figure 2: Illustration of pipeline of multi-phase prompt tuning LLM for context-aware NMT. Red lines illustrate the procedure of enhanced decoding phase.

Inter-sentence Context Encoding Phase. In the inter-sentence context encoding phase (Phase 1 in Figure 2), we first concatenate all sentences in $\mathcal{C}$ into a sequence, and then utilize the LLM to encode $\mathcal{C}$ by incorporating the trainable prompt:

$$H_{\mathcal{C}}^{1:L} = \text{LLM}(\mathcal{C}, \mathbf{P}_{\mathcal{C}}), \tag{1}$$

where $H_{\mathcal{C}}^{1:L} \in \mathbb{R}^{L\times|\mathcal{C}|\times d}$ is the sequence of activations for $\mathcal{C}$, $\mathbf{P}_{\mathcal{C}} \in \mathbb{R}^{L\times 2q\times d}$ is the current-phase trainable prompt, and q is a hyper-parameter for the length of the prompt. $\mathbf{P}_{\mathcal{C}}$ aims to adapt the LLM for better modeling the inter-sentence context. Same as basic deep prompting, at the l-th transformer block, we inject corresponding prompt in $\mathbf{P}_{\mathcal{C}}$ into encoding procedure of $\mathcal{C}$ as follows:

$$H_{\mathcal{C}}^{l} = \text{FFN}\left(\text{Multi-Attn}\left(\mathbf{K}_{\mathcal{C}}, \mathbf{V}_{\mathcal{C}}, \mathbf{Q}_{\mathcal{C}}\right)\right), \tag{2}$$

$$\mathbf{Q}_{\mathcal{C}} = H_{\mathcal{C}}^{l-1}, \tag{3}$$

$$\mathbf{K}_{\mathcal{C}} = [\mathbf{P}_{\mathcal{C}}[l,:q,:]; H_{\mathcal{C}}^{l-1}], \tag{4}$$

$$\mathbf{V}_{\mathcal{C}} = [\mathbf{P}_{\mathcal{C}}[l,q:,:]; H_{\mathcal{C}}^{l-1}], \tag{5}$$

where $H_{\mathcal{C}}^{l} \in \mathbb{R}^{|\mathcal{C}|\times d}$ is the output of the l-th transformer block. FFN and Multi-Attn are the feed-forward network sublayer and multi-head self-attention sublayer, respectively.[1] $[\cdot;\cdot]$ and $[\cdot:\cdot]$ are the concatenating and slicing operations, respectively.

Intra-sentence Context Encoding Phase. In the intra-sentence context encoding phase (Phase 2 in Figure 2), the LLM encodes the intra-sentence context S by conditioning on the past activations of the inter-sentence context $H_{\mathcal{C}}^{1:L}$ and trainable prompt:

$$H_{S}^{1:L} = \text{LLM}(S, H_{\mathcal{C}}^{1:L}, \mathbf{P}_{S}), \tag{6}$$

where $H_{S}^{1:L} \in \mathbb{R}^{L\times|S|\times d}$ is the sequence of activations for S, and $\mathbf{P}_{S} \in \mathbb{R}^{L\times 2q\times d}$ denotes current-phase prompt. Similarly, at the l-th transformer block, we incorporate $H_{\mathcal{C}}$ and $\mathbf{P}_{S}$ into the encoding procedure of S as follows:

$$H_{S}^{l} = \text{FFN}\left(\text{Multi-Attn}\left(\mathbf{K}_{S}, \mathbf{V}_{S}, \mathbf{Q}_{S}\right)\right), \tag{7}$$

$$\mathbf{Q}_{S} = H_{S}^{l-1}, \tag{8}$$

$$\mathbf{K}_{S} = [\mathbf{P}_{S}[l,:q,:]; H_{\mathcal{C}}^{l-1}; H_{S}^{l-1}], \tag{9}$$

$$\mathbf{V}_{S} = [\mathbf{P}_{S}[l,q:,:]; H_{\mathcal{C}}^{l-1}; H_{S}^{l-1}], \tag{10}$$

where H_{S}^{l} is output of the l-th transformer block, which fuses $H_{\mathcal{C}}^{l-1}$, the $l-1$ layer output of the inter-sentence context encoding.

Decoding Phase. In the decoding phase (Phase 3 in Figure 2), given the past activations H_S and trainable prompt, we call the LLM again to generate the hidden state for predicting the probability of the target sentence:

$$H_{T}^{1:L} = \text{LLM}(T, H_{S}^{1:L}, \mathbf{P}_{T}), \tag{11}$$

where $H_{T}^{1:L} \in \mathbb{R}^{L\times|T|\times d}$ is the sequence of activations for T, and $\mathbf{P}_{T} \in \mathbb{R}^{L\times 2q\times d}$ is current-phase prompt. Similarly, we inject S and $\mathbf{P}_{T}$ into the decoding procedure of T as follows:

$$H_{T}^{l} = \text{FFN}\left(\text{Multi-Attn}\left(\mathbf{K}_{T}, \mathbf{V}_{T}, \mathbf{Q}_{T}\right)\right), \tag{12}$$

$$\mathbf{Q}_{T} = H_{T}^{l-1}, \tag{13}$$

$$\mathbf{K}_{T} = [\mathbf{P}_{T}[l,:q,:]; H_{S}^{l-1}; H_{T}^{l-1}], \tag{14}$$

$$\mathbf{V}_{T} = [\mathbf{P}_{T}[l,q:,:]; H_{S}^{l-1}; H_{T}^{l-1}], \tag{15}$$

where $H_{T}^{l} \in \mathbb{R}^{|T|\times d}$ is the decoding state of the l-th transformer block. Finally, we refer the t-th decoding state as h_t^L (i.e., $H_T^L = h_t^L|_{t=1}^{|T|+1}$) which is used to predict the next token y_t:

$$p\left(y_t|S,\mathcal{C},y_{<t}\right) = \text{Softmax}\left(h_t^L W\right), \tag{16}$$

where $W \in \mathbb{R}^{d\times|\mathcal{V}|}$ is parameter of LLM-Head layer and $|\mathcal{V}|$ is the vocabulary size.

2.2 Enhanced Decoding Phase

As shown in Figure 2, both the inter-sentence context representation $H_{\mathcal{C}}^{1:L}$ and the intra-sentence context representation $H_{S}^{1:L}$ are used as keys and values when generating hidden states of next phase. Meanwhile, hidden states of decoding phase, i.e., $h_i^L|_{i=1}^{|T|}$ are used to predict next tokens. On the one hand, while the decoding hidden states incorporate both inter- and intra-sentence contexts, there is no explicit differentiation between the two when predicting next tokens. On the other hand, the inter-sentence context representation $H_{\mathcal{C}}^{1:L}$ and decoding hidden states $H_{T}^{1:L}$ are mediated by hidden states

[1] For simplicity, we omit the normalization and residual operations in this paper.

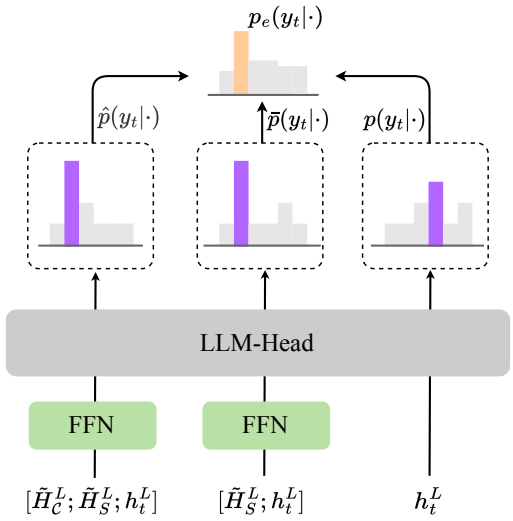

Figure 3: Illustration of the procedure of our proposed decoding-enhanced approach at the t-th decoding step of the decoding phase.

of phases 2, i.e., $H_S^{1:L}$. This may result in a *long-distance* issue such that the inter-sentence context are not properly aligned by target-side tokens.

Therefore, to address above two issues, we propose an enhanced decoding phase with an aim to more effectively utilize both the inter- and intra-sentence contexts. Inspired by Kuang et al. (2018), we move both the two types of inter- and intra-sentence contexts closer to target words to achieve a tight interaction between them. Specifically, we concatenate the decoding states with the two types of representations to predict the next target words. As shown in Figure 3, the enhanced next word prediction p_e is a combination of three distributions with different inputs:

$$\begin{aligned} p_e\left(y_t|S,\mathcal{C},y_{<t}\right) =& \lambda_1 \times \hat{p}\left(y_t|S,\mathcal{C},y_{<t}\right) \\ &+ \lambda_2 \times \bar{p}\left(y_t|S,\mathcal{C},y_{<t}\right) \\ &+ \left(1-\lambda_1-\lambda_2\right) \times p\left(y_t|S,\mathcal{C},y_{<t}\right), \end{aligned} \tag{17}$$

where λ_1 and λ_2 control the contribution of $\hat{p}\left(y_t|\cdot\right)$ and $\bar{p}\left(y_t|\cdot\right)$, respectively, which can be further computed as:

$$\hat{p}\left(y_t|S,\mathcal{C},y_{<t}\right) = \text{Softmax}\left(\hat{h}_t^L W\right), \tag{18}$$

$$\bar{p}\left(y_t|S,\mathcal{C},y_{<t}\right) = \text{Softmax}\left(\bar{h}_t^L W\right), \tag{19}$$

$$\hat{h}_t^L = \text{FFN}\left(\left[\tilde{H}_{\mathcal{C}}^L; \tilde{H}_S^L; h_t^L\right]\right), \tag{20}$$

$$\bar{h}_t^L = \text{FFN}\left(\left[\tilde{H}_S^L; h_t^L\right]\right), \tag{21}$$

where W is same as in Eq. 16, $\tilde{H}_S^L \in \mathbb{R}^d$ and $\tilde{H}_{\mathcal{C}}^L \in \mathbb{R}^d$ are the averaged H_S^L and $H_{\mathcal{C}}^L$ at token level, respectively.

2.3 Phase-aware Prompts

We emphasize the LLM needs to play various roles across three phases, and maintaining similar prompts across different phases may not be reasonable. Thus, we empower LLM to distinguish different phases by introducing a type embedding and a transfer layer[2] for these prompts:

$$\mathbf{P}_r = \left(\tanh\left(\mathbf{O}_r W_1\right)\right) W_2 + \text{TypeEmb}\left(r\right), \tag{22}$$

where $\mathbf{O}_r \in \mathbb{R}^{L\times 2q\times d}$ is randomly initialized prompt, $W_1, W_2 \in \mathbb{R}^{d\times d}$ are trainable parameters, and TypeEmb$(\cdot)$ is type embeddings layer of prompts. $r \in \{\mathcal{C}, S, T\}$ represents either phase 1, phase 2, or phase 3.

2.4 Training Objective

We employ the cross-entropy loss as the training objective of our model. Given a training instance $(\mathcal{C}, S, T)$, its training loss is defined as:

$$\mathcal{L}\left(\mathcal{C},S,T\right) = -\frac{1}{|T|}\sum_{t=1}^{|T|}\log p_e\left(y_t|S,\mathcal{C},y_{<t}\right). \tag{23}$$

Notably, the parameters in LLM, including W in Eq. 16, 18, 19, are frozen during training.

3 Experimentation

We build our approach upon two open-source LLMs, namely, `llama-2-7b`[3] and `bloomz-7b1-mt`[4]. We verify the effectiveness of our proposed approach on five translation tasks, including {Chinese (ZH), French (FR), German (DE), Spanish (ES), Russian (RU)}→English (EN).

3.1 Experimental Settings

Datasets and Preprocessing. The corpus of all translation tasks is extracted from `New-Comentary-v18`. See Appendix A for splitting and statistics of the training set, valid set, and test set. We use the tokenizer of foundation models to process the input data and no any other preprocessing is performed.

[2]Different from the multi-layer perceptron (MLPs) used for reparameterization, our transfer layer is shared-parameter for all prompts. Thus, there are fewer trainable parameters during the training of our model. We compare the number of trainable parameters among different tuning methods in Table 3 and analyze the effect of the transfer layer in Appendix D.

[3]https://huggingface.co/meta-llama/Llama-2-7b-hf

[4]https://huggingface.co/bigscience/bloomz-7b1-mt

Model	ZH→EN		FR→EN		DE→EN		ES→EN		RU→EN	
	BLEU	*COMET*	*BLEU*	*COMET*	*BLEU*	*COMET*	*BLEU*	*COMET*	*BLEU*	*COMET*
Trans.	29.86	0.8406	38.53	0.8545	41.44	0.8682	48.74	0.8783	32.25	0.8169
`llama-2-7b` as foundation model										
MT-LoRA	27.43	0.8511	38.18	0.8647	40.96	0.8712	47.52	0.8733	33.00	0.8311
MT-PT	31.32	0.8565	41.92	0.8675	43.56	0.8752	51.32	0.8819	35.46	0.8333
CMT-PT	31.13	0.8387	42.01	0.8699	43.11	0.8762	51.66	0.8823	35.91	0.8396
MPT	*33.21	0.8645	†43.11	0.8744	*43.88	0.8824	†52.01	0.8913	†36.49	**0.8456**
DeMPT	***33.89**	**0.8658**	**†43.71**	**0.8816**	***44.69**	**0.8899**	**†53.10**	**0.8979**	**†36.55**	0.8438
`bloomz-7b1-mt` as foundation model										
MT-LoRA	25.79	0.8466	35.67	0.8601	35.17	0.8522	46.32	0.8644	28.01	0.8012
MT-PT	30.99	0.8520	40.49	0.8661	37.76	0.8579	50.68	0.8823	30.27	0.8106
CMT-PT	30.82	0.8504	40.31	0.8639	38.01	0.8601	50.26	0.8832	29.80	0.8108
MPT	*31.81	0.8601	*41.11	0.8766	†38.99	0.8669	*51.33	0.8910	*30.99	0.8201
DeMPT	***32.46**	**0.8649**	***41.92**	**0.8790**	**†40.06**	**0.8703**	***52.25**	**0.8990**	***31.79**	**0.8253**

Table 1: Results of different systems on sacreBLEU and COMET metrics. **DeMPT/MPT** is our proposed Multi-phase Prompt Tuning approach *with/without* Decoding-enhanced strategy (in Sec. 2.2). Scores with **bold** indicate the best performance. * or † indicates the gains are statistically significant over MT-PT or CMT-PT with p<0.01 (Koehn, 2004).

Model	ZH→	FR→	DE→	ES→	RU→
Trans.	47.63	54.41	58.29	62.52	48.79
`llama-2-7b` as foundation model					
MT-LoRA	44.83	54.52	57.72	62.18	49.06
MT-PT	49.49	57.87	60.89	65.02	52.59
CMT-PT	49.53	58.27	61.23	65.89	53.34
MPT	51.56	59.56	62.15	67.14	54.18
DeMPT	**52.68**	**60.33**	**63.11**	**67.95**	**54.34**
`bloomz-7b1-mt` as foundation model					
MT-LoRA	43.23	51.82	51.12	61.77	43.29
MT-PT	49.48	56.81	55.40	64.71	46.14
CMT-PT	49.61	57.05	55.81	65.12	46.09
MPT	50.22	57.93	56.69	66.25	47.29
DeMPT	**50.62**	**58.30**	**57.34**	**67.12**	**48.00**

Table 2: Results of different systems on BlonDe metric.

Baselines. We compare our approach against four baselines:

- **Transformer** (Trans.): It is an encoder-decoder Transformer-base model (Vaswani et al., 2017) that is trained from scratch.
- **MT-LoRA**: It is a tuned LLM adapted to NMT task via the tuning method of Low-Rank Adaptation (Hu et al., 2022), which makes large-scale pre-training models adapt to a new task by injecting a trainable rank decomposition matrice into each layer of the Transformer architecture.
- **MT-PT**: It is a tuned LLM adapted to NMT task via the deep prompt tuning with MLPs reparameterization,[5] which only tunes continuous prompts with a frozen language model.
- **CMT-PT**: Similar to MT-PT, it is also a tuned LLM via the deep prompt tuning with MLPs reparameterization. Unlike MT-PT, it utilizes inter-sentence context within the concatenation strategy, as depicted in Figure 1 (a).

Among them, Transformer, MT-LoRA, and MT-PT are *context-agnostic* systems while CMT-PT is a *context-aware* system. For a fair comparison, we ensure that all context-aware systems, including CMT-PT, MPT, and DeMPT, incorporate identical inter-sentence context.

Model Setting and Training. For the Transformer model, we implement it upon Fairseq (Ott et al., 2019). For MT-LoRA models, we set the rank of trainable matrices as 16 which performs best in our preliminary experiment. For all MT-PT models, CMT-PT models, and our models, we set the prompt length q as 64. For the incorporation

[5]We attempt to remove reparameterization but experience a significant decline in performance.

of inter-sentence context in CMT-PT models and our models, we consider a dynamic z, in which the total tokens are no more than 256. In enhanced decoding, we consider the three next word predictions to be equally important by setting both λ_1 and λ_2 to 1/3. More details of training are provided in Appendix B.

Evaluation. We use sacreBLEU (accuracy-related metric)[6] (Post, 2018), COMET (semantics-related metric) with the `wmt22-comet-da` model[7] (Rei et al., 2020), and BlonDe (discourse-related metric) (Jiang et al., 2022) as the evaluation metrics.

3.2 Experimental Results

The main experimental results are presented in Tables 1 and 2. Additionally, a comparison of the number of trainable parameters is presented in Table 3 across different tuning methods. From these results we have the following observations:

- The encoder-decoder Transformer model (Trans.) performs better than the LLMs with LoRA tuning in most translation directions in BLEU score. For example, when utilizing `llama-2-7b` as the foundation model, Transformer surpasses MT-LoRA an average of 0.75 BLEU score across all translation tasks. However, MT-LoRA model outperforms Trans. in terms of COMET, suggesting that translations from LLMs may align more closely with human preferences.

- The MT-PT model presents superior performance compared to the MT-LoRA model in terms of BLEU, COMET, and BlonDe. Taking `bloomz-7b1-mt` as the foundation model, the MT-PT model outperforms the MT-LoRA model by an average of 3.84 BLEU score, 3.51 BlonDe score, and 0.0047 COMET score. Nevertheless, the MT-PT model sacrifices efficiency for performance, introducing more trainable parameters (13.87% vs. 0.12%).

- Leveraging the inter-sentence context is helpful in alleviating discourse-related issues. For example, with `bloomz-7b1-mt` used as the foundation model, the CMT-PT model, despite underperforming in BLEU and COMET compared to the MT-PT model, excels in discourse-related BlonDe scores (averaging 57.66 vs. 57.17).

- Our MPT/DeMPT model outperforms all baselines across all translation tasks. For example, when using `llama-2-7b` as the foundation model, our MPT model achieves an average gain of 1.62/1.45/2.03 in BLEU/COMET/BlonDe score compared to the CMT-PT model. Furthermore, our decoding-enhance strategy enhances the capacity of LLMs in context-aware NMT, with DeMPT outperforming MPT in BLEU/COMET/BlonDe score (averaging 42.39/0.8758/59.68 vs. 41.74/0.8716/58.91).

- The model built upon `llama-2-7b` as the foundation model outperforms the one using `bloomz-7b1-mt`, suggesting that `llama-2-7b` serves as a more robust foundation model for translation tasks.

	MT-LoRA	MT-PT/CMT-PT	DeMPT
Trainable Para.	0.12%	13.87%	3.11%

Table 3: Proportion of trainable parameters against total parameters for different tuning methods.

4 Discussion

In this section, we use `bloomz-7b1-mt` as the foundation model to discuss and analyze our approach. See Appendix C∼E for further discussions.

4.1 Effect of Length of Inter-sentence Context

For efficient training, we define the inter-sentence context in Section 2 as previous sentences with a total tokens not exceeding 256. We are curious about the potential impact of inter-sentence length on the performance of our approach. Consequently, we extend the inter-sentence context length from 256 to 1024 and assess the performance of our approach in the ZH→EN task.

Figure 4 shows the performance trend of the CMT-PT model and our DeMPT model. As the length of the inter-sentence context increases, both models exhibit a slight enhancement in both BLEU and BlonDe scores. Interestingly, our model with a 256-token inter-sentence context outperforms the CMT-PT model with a 1024-token inter-sentence context in both BLEU and BlonDe scores. This

[6]Signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`

[7]https://github.com/Unbabel/COMET

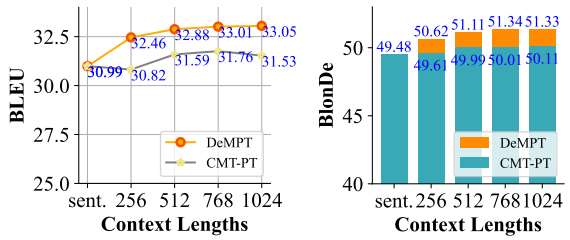

Figure 4: Performance of CMT-PT and our DeMPT on ZH→EN test set when using different inter-sentence context lengths.

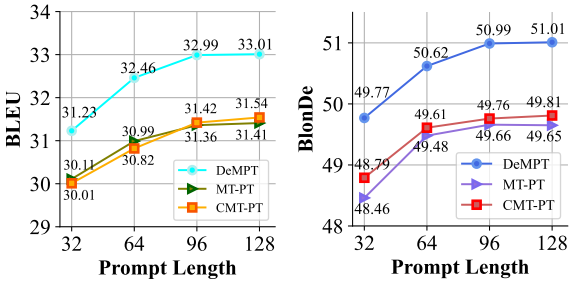

Figure 5: Performance of MT-PT, CMT-PT, and our DeMPT on ZH→EN test set when using different lengths of the trainable prompts.

further suggests the effectiveness of our approach in harnessing the capabilities of LLMs for context-aware NMT compared to the concatenation strategy.

4.2 Effect of Prompt Length

As our approach is implemented based on deep prompt tuning, next we compare the impact of the trainable prompt length for MT-PT, CMT-PT, and our DeMPT.

Figure 5 shows the performance curves when increasing the prompt length from 32 to 128. We observe that increased prompt length tends to enhance performance for both BLEU and BlonDe, yet the gains exhibit diminishing returns. This finding is consistent with that in Li and Liang (2021); Lester et al. (2021); Tan et al. (2022). We also observe that DeMPT with a prompt length of 64 outperforms both MT-PT and CMT-PT with a prompt length of 128 on both metrics, suggesting the superiority of our approach over the concatenation strategy in enhancing LLMs' capacity for context-aware NMT.

4.3 Comparison of Inference Speed

Table 4 compares the inference speed of different models on ZH→EN translation task. Our MPT and DeMPT models, dividing the context-aware NMT process into three separate phases, demonstrates comparable inference speed to the single-phase MT-PT and CMT-PT models, with only a marginal drop of 0.02 seconds per sentence in decoding. This illustrates the efficiency of our approach without introducing significant computational overhead.

Model	Speed	BLEU
MT-PT	0.75 *sec/sent.*	30.99
CMT-PT	0.77 *sec/sent.*	30.82
MPT	0.78 *sec/sent.*	31.81
DeMPT	0.79 *sec/sent.*	32.46

Table 4: Comparison of inference speed on ZH→EN translation task. Speed is measured on the test set using 4 GPUs. *sec/sent.* means seconds spent for decoding each sentence. Note that the reparameterization is not needed during inference (Li and Liang, 2021).

Model	deixis	lex.c	ell.infl	ell.VP	Avg.
MT-PT	50.0	45.7	53.0	28.6	44.3
CMT-PT	**80.2**	46.1	74.3	75.3	68.9
DeMPT	80.1	**55.7**	**75.9**	**79.3**	**72.7**

Table 5: Accuracy [%] of translation prediction for four discourse phenomena on the English → Russian contrastive test set.

4.4 Performance on Contrastive Test Set

We evaluate the models' ability to resolve discourse inconsistencies using the contrastive test set proposed by (Voita et al., 2019a), which focuses on four discourse phenomena such as deixis, lexicon consistency (lex.c), ellipsis inflection (ell.infl), and verb phrase ellipsis (ell.VP) in English→Russian translation. Within the test set, each instance comprises a positive translation and several negative ones that vary by only one specific word. The purpose of the contrastive test set is to assess whether a model is more inclined to generate a correct translation as opposed to incorrect variations.

Table 5 lists the accuracy of translation prediction on the contrastive test set for MT-PT, CMT-PT and DeMPT. Compared to the context-agnostic MT-PT model, both context-aware CMT-PT and DeMPT models show substantial improvements across the four discourse phenomena. Additionally, DeMPT demonstrates the best performance, surpassing CMT-PT by an average accuracy margin of 3.8.

Model	Score_1	Score_2	Average
CMT-PT	79.00	80.17	79.59
DeMPT	**86.17** (+7.17)	**87.30** (+7.13)	**86.73** (+7.14)

Table 6: Human DA scores for CMT-PT and DeMPT on ZH→EN translation task.

4.5 Human Evaluation

We use the Direct Assessment (DA) method (Graham et al., 2017) to manually assess the quality of translations generated by DeMPT and CMT-PT. In this assessment, human evaluators compare the meaning of the MT output with a human-produced reference translation, working within the same language.

Specifically, we randomly select 5 documents with a total of 200 groups of sentences from the ZH→EN test set. To avoid potential bias in evaluation, we recruit 6 professional translators and ensure each translation from DeMPT or CMT-PT is scored twice by two translators. Table 6 shows the DA scores for CMT-PT and DeMPT. Our DeMPT outperforms CMT-PT by 7.14 DA score, providing strong evidence for the effectiveness of our approach. Further details and results regarding the DA can be found in Appendix C.

5 Related Work

Large Language Models for Context-aware Machine Translation. While traditional context-aware neural machine translation (NMT) has seen considerable progress in recent years (Jean et al., 2017; Wang et al., 2017; Voita et al., 2018; Maruf et al., 2019; Kang et al., 2020; Bao et al., 2021; Sun et al., 2022; Bao et al., 2023), the effective integration of large language models (LLMs) to model inter-sentence context and enhance context-aware translation remains an area of limited exploration. Existing studies mainly focus on the assessment of LLMs' ability in discourse modeling. For example, Wang et al. (2023) approach context-aware NMT as a task involving long sequence generation, employing a concatenation strategy, and conduct comprehensive evaluations of LLMs such as ChatGPT and GPT-4. Their focus includes the impact of context-aware prompts, comparisons with translation models, and an in-depth analysis of discourse modeling ability. Similarly, Karpinska and Iyyer (2023) engage professional translators to evaluate LLMs' capacity in context-aware NMT. In contrast, Wu et al. (2024) compare the effectiveness of various parameter-efficient fine-tuning methods on moderately-sized LLMs for context-aware NMT. Besides, Wu and Hu (2023) explore the prompt engineering with GPT language models specifically for document-level (context-aware) MT while Li et al. (2024) experiment with combining sentence-level and document-level translation instructions of varying lengths to fine-tune LLMs.

Prompt Tuning for Large Language Model. Liu et al. (2021) and Li and Liang (2021) propose to make LLMs adapt to various tasks by adding trainable prompts (also called continuous prompts) to the original input sequences. In this paradigm, only the continuous prompts are updated during training. Liu et al. (2022) further introduce deep prompt tuning, extending the idea by inserting trainable prompts into all layers of LLMs, rather than just the embedding layer. While these approaches lay the groundwork for a general framework, our focus lies in augmenting the performance of LLMs specifically for inter-sentence context modeling in context-aware NMT. Notably related, Tan et al. (2022) propose a multi-phase tuning approach to enhance the sentence-level translation performance of a multilingual GPT. However, the exploration of effective LLM tuning for addressing discourse-related challenges in the context-aware NMT domain remains underdeveloped.

6 Conclusion

In this paper, we have examined the hypothesis that it is crucial to differentially model and leverage inter-sentence context and intra-sentence context when adapting LLMs to context-aware NMT. This stems from our observation that intra-sentence context exhibits a stronger correlation with the target sentence compared to inter-sentence context, owing to its richer parallel semantic information. To this end, we have proposed a novel decoding-enhanced multi-phase prompt tuning (DeMPT) approach to make LLMs aware of the differences between inter- and intra-sentence contexts, and further improve LLMs' capacity in discourse modeling. We have evaluated our approach using two foundation models and present experimental results across five translation directions. Experimental results and discussions have demonstrated a significant enhancement in the performance of LLMs in context-aware NMT, manifesting as improved translation accuracy and a reduction in discourse-related issues.

Limitations

Owing to resource limitations, our work is restricted to moderate-scale LLMs, specifically those with 7 billion parameters, and a confined window size of inter-sentence context. It is imperative to acknowledge that the results of our research may differ when employing larger models and an extended window size for inter-sentence context. We acknowledge these limitations and consider them as avenues for future exploration.

References

Guangsheng Bao, Zhiyang Teng, and Yue Zhang. 2023. Target-side augmentation for document-level machine translation. In *Proceedings of ACL*, pages 10725–10742.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of ACL*, pages 3442–3455.

BigScience. 2022. Bloom: A 176b-parameter open-access multilingual language model. *Computing Research Repository*, arXiv:2211.05100.

Google. 2022. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra, and Carolina Scarton. 2017. Improving evaluation of document-level machine translation quality estimation. In *Proceedings of EACL*, pages 356–361.

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 *Proceedings of ICLR*.

Xinyu Hu and Xiaojun Wan. 2023. Exploring discourse structure in document-level machine translation. In *Proceedings of EMNLP*, pages 13889–13902.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *Computing Research Repository*, arXiv:1704.05135.

Yuchen Eleanor Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Jian Yang, Haoyang Huang, Rico Sennrich, Ryan Cotterell, Mrinmaya Sachan, and Ming Zhou. 2022. BlonDe: An automatic evaluation metric for document-level machine translation. In *Proceedings of NAACL*, pages 1550–1565, Seattle, United States.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of EMNLP*, pages 2242–2254.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. In *Proceedings of WMT*, pages 419–451.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.

Shaohui Kuang, Junhui Li, António Branco, Weihua Luo, and Deyi Xiong. 2018. Attention focusing for neural machine translation by bridging source and target embeddings. In *Proceedings of ACL*, pages 1767–1776.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP*, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL-IJCNLP*, pages 4582–4597, Online.

Yachao Li, Junhui Li, Jing Jiang, Shimin Tao, Hao Yang, and Min Zhang. 2023. P-Transformer: Towards Better Document-to-Document Neural Machine Translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3859–3870.

Yachao Li, Junhui Li, Jing Jiang, and Min Zhang. 2024. Enhancing document-level translation of large language model via translation mixed-instructions. *Computing Research Repository*, arXiv:2401.08088.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of ACL*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *Computing Research Repository*, arXiv:2103.10385.

Xinglin Lyu, Junhui Li, Zhengxian Gong, and Min Zhang. 2021. Encouraging lexical translation consistency for document-level neural machine translation. In *Proceedings of EMNLP*, pages 3265–3277.

Xinglin Lyu, Junhui Li, Shimin Tao, Hao Yang, and Min Zhang. 2022. Modeling consistency preference via lexical chains for document-level neural machine translation. In *Proceedings of EMNLP*, pages 6312–6326.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of NAACL*, pages 3092–3102.

MetaAI. 2023a. Llama 2: Open foundation and fine-tuned chat models. *Computing Research Repository*, arXiv:2307.09288.

MetaAI. 2023b. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of EMNLP*, pages 2947–2954.

OpenAI. 2023. Gpt-4 technical report. *Computing Research Repository*, arXiv:2303.08774.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of WMT*, pages 186–191.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: Proceedings of High Performance Computing, Networking, Storage and Analysis*, pages 1–16.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of EMNLP*, pages 2685–2702.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of ACL*, pages 3537–3548.

Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. 2022. MSP: Multi-stage prompting for making pre-trained language models better translators. In *Proceedings of ACL*, pages 6131–6142.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of EMNLP-IJCNLP*, pages 877–886.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of ACL*, pages 1198–1212.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of ACL*, pages 1264–1274.

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 *Proceedings of EMNLP*, pages 16646–16661.

Longyue Wang, Zhaopeng Tu, Xing Wang, and Shuming Shi. 2019. One model to learn both: Zero pronoun prediction and translation. In *Proceedings of EMNLP-IJCNLP*, pages 921–930.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of EMNLP*, pages 2826–2831.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari. 2024. Adapting large language models for document-level machine translation. *Computing Research Repository*, arXiv:2401.06468.

Yangjian Wu and Gang Hu. 2023. Exploring prompt engineering with GPT language models for document-level machine translation: Insights and findings. In *Proceedings of the Eighth Conference on Machine Translation*, pages 166–169.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of EMNLP*, pages 533–542.

A Datasets

Statistics and Splitting of Datasets. We provide the detailed statistic in Table 7. For all translation tasks, we randomly select 80% document pairs from the corpus as the training set. Both the test set and validation set include 150 document pairs each, randomly sampled from the remaining 20% of document pairs in the corpus. Regarding sentence preprocessing across all datasets, we segment the sentences with the tokenizer from the respective foundation model. No additional preprocessing steps are performed. Datasets are downloaded from `https://data.statmt.org/news-commentary/v18`.

B Training Details

For all Transformer NMT models, we use the transformer-base setting as in Vaswani et al. (2017), where the learning rate is set to 1e-4. The Transformer NMT models are trained on 4× NVIDIA V100 32GB GPUs with a batch size of 4096. For the models with prompt tuning in Section 3, including MT-PT, CMT-PT, and our MPT and DeMPT models, the length of the trainable prompt is set as 64. During both training and inference, the model generates only the current target sentence, operating in a many-to-one translation mode.. For all fine-tuning models in this paper, we set the training epoch to 4, and the warm-up rate to 0.1. We use

Dataset	ZH→EN		FR→EN		DE→EN		ES→EN		RU→EN	
	#Doc	#Sent	#Doc	#Sent	#Doc	#Sent	#Doc	#Sent	#Doc	#Sent
Training	8,622	342,495	7,915	310,489	8,417	333,201	9,677	378,281	7,255	272,100
Validation	150	6,061	150	5,890	150	5,866	150	5,782	150	5,691
Test	150	5,747	150	5,795	150	5,967	150	5,819	150	5,619

Table 7: Statistics of training, validation, and test sets for five translation tasks. #Doc and #Sent denote the numbers of *Document* and *Sentence*, respectively.

the log learning rate decay strategy with a maximum learning rate of 5e-5. We collate a mini-batch by counting the total tokens inside the batch and set the batch size as 4096. All fine-tuning models are trained on 4 × NVIDIA A800 GPUs with Deespeed Zero 2 offload setting (Rajbhandari et al., 2020).[8]

C Details of Human Evaluation

Criterion and Recruitment. Given a source sentence, its translation from MT (i.e., CMT-PT and our DeMPT), and its human-produced reference translation, the evaluators are asked to give a score ranging from 0 to 100. Figure 6 presents the detailed criterion of scoring. We recruit evaluators from professional translators with at least five years of experience in translation.

Statistics of Translation Errors. We manually count the number of bad cases from our DeMPT model. The bad cases fall into two categories: (1) the DA score is 60 or lower; (2) the DA score is lower than that of the translation from CMT-PT. The main types of the bad cases are **Mistranslation** (`Mis.`), **Unnoticed Omission** (`UO`), **Inappropriate Expression** (`IE`), and **Grammatical Error** (`GE`). We present detailed statistics in Table 8. The statistics indicate the bad cases mainly come from Mistranslation and Unnoticed Omission. Meanwhile, our DeMPT model outperforms the CMT-PT model in 86.5% DA cases.

Case Study. We present a case in Figure 7 to illustrate how our DeMPT model outperforms the CMT-PT model. In this case, we compare the translations of two consecutive sentences from our model and the CMT-PT model. First, we notice that the CMT-PT model translates the source word 美国 in the two sentences into *US* and *America*, respectively. However, our model **consistently** translates them into *US*. Second, our model uses *for its part*, a phase with more **coherent preference**, as the translation of 同时 , instead of *At the same time* adopted in the translation from the CMT-PT model. Both of them demonstrate the superiority of our proposed approach in discourse modeling.

[8] https://github.com/microsoft/DeepSpeed

Group	Type of Bad Case				
	`Mis.`	`UO`	`IE`	`GE`	*Total (Perc.)*
1	6	3	1	2	12 (**6.0%**)
2	9	7	6	5	27 (**13.5%**)

Table 8: Statistics of bad cases from our DeMPT model on ZH→EN translation task. *Perc.* denotes the percentage of bad cases against the total of DA cases.

D Effect of Transfer Layer and Type Embedding

As in Eq. 22 within Section 2.3, we introduce two sublayers: a non-linear transfer sublayer and a type embedding sublayer for the trainable prompt in each phase. This design enhances the awareness of LLMs regarding the distinctions in inputs across the three tuning phases, allowing them to adapt to specific roles at each phase. We investigate the effect of these two sublayers.

As shown in Table 9, our observations reveal that the transfer sublayer holds greater importance than the type embedding sublayer. Removing either the non-linear transfer sublayer (*w/o* Transfer.) or the type embedding sublayer (*w/o* Embed.) results in a performance drop of 0.84/0.0048/0.39 or 0.45/0.0036/0.007 in BLEU/COMET/BlonDe metrics.

E Effect of Inter-sentence Context

We implement the context-agnostic (sentence-level) DeMPT system to analyze the effect of the inter-sentence context. More specifically, we replace the input of LLMs in the inter-sentence context encoding phase with the intra-sentence context. In

Score	Criterion
0-20	The translation is completely incorrect and unclear, with only a few words or phrases being correct. It is totally unreadable and difficult to understand.
21-40	The translation has very little semantic similarity to the source sentence, with key information missing or incorrect. It has numerous unnatural and unfluent expressions and grammatical errors.
41-60	The translation can express part of the key semantics but has many non-key semantic errors. It lacks fluency and idiomaticity.
61-80	The translation can express the key semantics but has some non-key information errors and significant grammatical errors. It lacks idiomaticity.
81-100	The translation can express the semantics of the source sentence with only a few non-key information errors and minor grammatical errors. It is fluent and idiomatic.

Figure 6: Scoring criterion for Direct Assessment. We group the score into five ranges, i.e., 0-20, 21-40, 41-60, 61-80, 81-100.

Model	BLEU	COMET	BlonDe
MT-PT	30.99	0.8520	49.48
CMT-PT	30.82	0.8504	49.61
DeMPT	32.46	0.8649	50.62
w/o Transfer.	31.62	0.8601	50.23
w/o Embed.	32.01	0.8613	50.55
w/o CTX.	31.98	0.8593	49.89

Table 9: Comparison of performances of the DeMPT variants on ZH→EN test set. *w/o* Trans. or *w/o* Embed. denotes the variant without the non-linear transfer sublayer or type embedding sublayer in Eq. 22. *w/o* CTX. means the inter-sentence context is not available, i.e., context-agnostic DeMPT system.

other words, we encode the intra-sentence context twice to keep the multi-phase tuning strategy in DeMPT while making the inter-sentence context unavailable.

As shown in the last row of Table 9 (i.e., *w/o* CTX), we find that the inter-sentence context is crucial for the alleviation of discourse-related issues. The BlonDe score drops by 0.73 when the inter-sentence context is unavailable. Meanwhile, our DeMPT also significantly improves the performance of LLMs in context-agnostic MT, e.g., + 0.99 BLEU score and + 0.0073 COMET score compared to the MT-PT model.

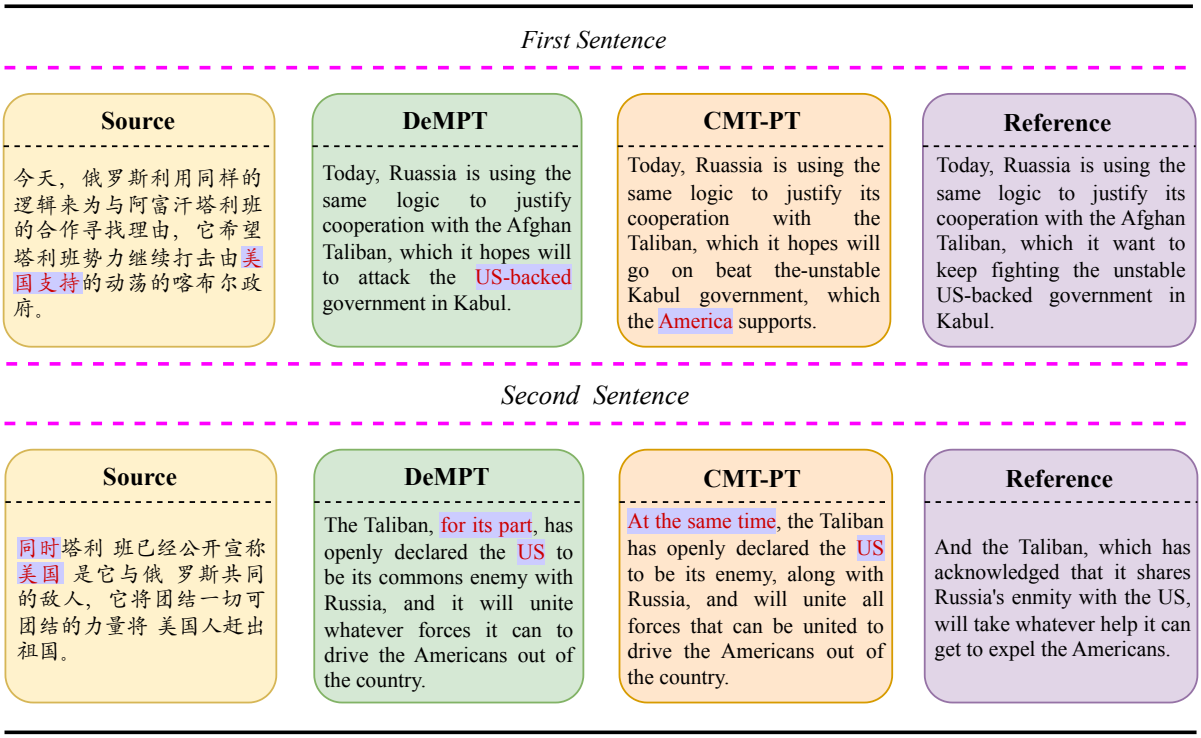

Figure 7: A case study for the CMT-PT model and our DeMPT model on ZH→EN translation task.