LCS: A Language Converter Strategy for Zero-Shot Neural Machine Translation

Zengkui Sun[1]*, Yijin Liu[2], Fandong Meng[2], Jinan Xu[1], Yufeng Chen[1]† and Jie Zhou[2]
[1]Beijing Jiaotong University, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
{zengksun,jaxu,chenyf}@bjtu.edu.cn
{yijinliu,fandongmeng,withtomzhou}@tencent.com

Abstract

Multilingual neural machine translation models generally distinguish translation directions by the language tag (LT) in front of the source or target sentences. However, current LT strategies cannot indicate the desired target language as expected on zero-shot translation, *i.e.*, the *off-target* issue. Our analysis reveals that the indication of the target language is sensitive to the placement of the target LT. For example, when placing the target LT on the decoder side, the indication would rapidly degrade along with decoding steps, while placing the target LT on the encoder side would lead to copying or paraphrasing the source input. To address the above issues, we propose a simple yet effective strategy named **L**anguage **C**onverter **S**trategy (**LCS**). By introducing the target language embedding into the top encoder layers, LCS mitigates confusion in the encoder and ensures stable language indication for the decoder. Experimental results on MultiUN, TED, and OPUS-100 datasets demonstrate that LCS could significantly mitigate the *off-target* issue, with language accuracy up to 95.28%, 96.21%, and 85.35% meanwhile outperforming the vanilla LT strategy by 3.07, 3,3, and 7.93 BLEU scores on zero-shot translation, respectively.

LT Strategies	**Translation Pair (De→Fr)**	
Source	Wie sind sie durch die Sicherheitskontrolle gekommen?	*(German)*
Reference	Comment avez-vous passé le contrôle de sécurité?	*(French)*
T-Enc	Wie sind sie durch die Sicherheitskontrolle gekommen?	*(German)*
S-Enc-T-Dec	How'd they get through the security?	*(English)*
LCS (ours)	Comment sont-elles venues par les contrôles de sécurité?	*(French)*

Table 1: Translation results of the same translation pair (German→French) with three LT strategies[1]. T-Enc leads to mistranslating into the undesired German sentence and S-Enc-T-Dec leads to mistranslating into the English sentence, while LCS accurately leads to translating into the French sentence.

1 Introduction

Multilingual Neural Machine Translation (MNMT) aims to build a unified model to support the translation between any language pairs (Dabre et al., 2020; Zhang et al., 2021; Fan et al., 2021). One main challenge is the indication of the translation direction. Pioneers (Dong et al., 2015; Luong et al., 2015; Firat et al., 2016a; Lu et al., 2018) utilize the language-specific encoder or decoder to distinguish source or target language. To simplify the architecture, Firat et al. (2016b) and Johnson et al. (2017) propose the language tag (LT) strategy, which places an artificial language tag in front of the source input sentence, to indicate the desired target language without modification on the vanilla NMT architecture and training objective. Due to its simplicity and efficiency, the LT strategy has become the *de facto* strategy to build unified MNMT models (Johnson et al., 2017; Dabre et al., 2020; Zhang et al., 2020; Fan et al., 2021), even other unified models (Zhang et al., 2023; Wang et al., 2022b; Liang et al., 2023).

With the LT strategy, the MNMT models theoretically support many-to-many translation, and even zero-shot translation (Johnson et al., 2017; Dabre et al., 2020; Gao et al., 2023). However, in practice, the MNMT models frequently mistranslate the source language to the wrong target language on zero-shot translation, referred to as the *off-target* issue (Zhang et al., 2020). Tab.1 shows an example of zero-shot translation from German to French, in which T-Enc and S-Enc-T-Dec[1] are commonly used LT strategies (Johnson et al., 2017; Fan et al., 2021; Wu et al., 2021). In this case, neither of both strategies could help MNMT models translate the sentence into the correct target language. The T-Enc strategy leads to the *To-Source* issue (*i.e.*, paraphrase or copy the input of source language),

[*] Work was done when Zengkui Sun was an intern at Pattern Recognition Center, WeChat AI, Tencent Inc, China.
[†] Yufeng Chen is the corresponding author.

[1] S / T represents the source / target LT, and Enc / Dec represents the LT placed on the Encoder / Decoder side. Tab.2 shows more detailed examples.

Findings of the Association for Computational Linguistics ACL 2024, pages 9201–9214
August 11-16, 2024 ©2024 Association for Computational Linguistics

while S-Enc-T-Dec leads to the *To-English* issue (*i.e.*, translate into English). For the *To-Source* issue, prior studies (Wang et al., 2022c) suggest that the data noises make this issue. However, our experiments display that the *To-Source* issue still exists after data denoise and the different language tag strategies result in different error distribution on both issues (§3.2). And for the *To-English* issue, we suspect it comes from the inadequate language indication of the target language and the most-common language (*i.e.*, English) is model preferred.

To further investigate the above issues, we conduct experiments to explore the indication of the target language in various LT strategies. And we summarize some preliminary findings:

- Placing the target language tag on the encoder side yields a more stable indication, while placing it on the decoder side delivers a decreasing indication throughout decoding steps and results in the *To-English* issue (§3.3).
- The encoder tends to convert the states to be target language-specific on the top layers, and placing the target LT at the top encoder layers could mitigate the *To-Source* issue (§3.4).
- Unfortunately, mainstream LT strategies suffer at least one issue between the *To-Source* and *To-English* in different degrees, even data denoising or placing the target language tag at the top encoder layers (§3.2 & §3.4).

On these grounds, we propose a simple yet effective strategy, **L**anguage **C**onverter **S**trategy (**LCS**), to address the above issues. Specifically, we first split the encoder layers and introduce the target language information to the deeper layers, which is named Language Converter by us. Compared to the mere placement of language tag, we supply the target language embedding, which contains language-specific features (Bjerva et al., 2019; Oncevay et al., 2020; Jin and Xiong, 2022), into each input state in each layer of the Language Converter layers. In this manner, LCS could provide stable and sufficient target language indication for MNMT model, avoiding both *To-Source* and *To-English* issues.

Experimentally, LCS could significantly mitigate the *off-target* issue and boost the performance of zero-shot translation. Specifically, compared to the most widely-used strategy, *i.e.*, the T-Enc strategy, LCS effectively mitigates the *off-target* issue by improving language accuracy up to 95.28% (*+2.7%*), 96.21% (*+1.71%*), 85.35% (*+40.97%*), and 86.67% (*+5.03%*) on zero-shot translation of MultiUN, TED, OPUS-100 (noise and denoised) datasets, respectively. Furthermore, LCS outperforms the T-Enc strategy by 3.07, 3.3, 7.93, and 2.93 BLEU scores improvements on zero-shot translation of these datasets, respectively. Meanwhile, LCS maintains the performance of the supervised translation and performs well on the noise data. Moreover, LCS is well compatible with other approaches, *e.g.*, denoising-encoder (Wang et al., 2021), contrastive learning (Pan et al., 2021), LEE (Jin and Xiong, 2022), and mBART (Liu et al., 2020), and yields further improvements when applied to them.

The main contributions of this paper can be summarized as follows[2]:

- We take the analysis of the *To-Source* and *To-English* issues in the *off-target* issue, and explore their causes in terms of the language representation variation of the encoder and language generation.
- We propose a simple yet effective strategy, LCS, to address the *off-target* issue and further improve zero-shot translation quality, without introducing extra parameters.
- Experimental results demonstrate that LCS could significantly mitigate the *off-target* issue and further boost the performance of zero-shot translation, with strong compatibility.

2 Background

2.1 Multilingual Neural Machine Translation

In bilingual neural machine translation (NMT), given a source sentence with n tokens $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and its target sentence with m tokens $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$, and NMT models are generally optimized by the cross-entropy loss:

$$\mathcal{L}_{\text{NMT}}(\theta) = -\sum_{j=1}^{m} \log p(y_j|\mathbf{y}_{<j}, \mathbf{x}; \theta), \quad (1)$$

where j is the index of each decoding step, $\mathbf{y}_{<j}$ is the target-side previous context for y_j, and θ represents the model parameter.

While in MNMT, for the language pair $(\mathbf{x}^s, \mathbf{y}^t)$ where s and t represent the source language s and the target language t, and $\mathbf{t}$ denotes the target language tag, the MNMT models are generally opti-

[2] Codes are released at `https://github.com/Acerkoo/LCS`.

LT Strategies	Example (En → De)
T-Enc (Johnson et al., 2017)	<de> Hello, how are you? Hallo, wie geht's?
S-Enc-T-Dec (Fan et al., 2021)	<en> Hello, how are you? <de> Hallo, wie geht's?
ST-Enc (Xue et al., 2021)	<en> <de> Hello, how are you? Hallo, wie geht's?
ST-Enc-T-Dec (ElNokrashy et al., 2022)	<en> <de> Hello, how are you? <de> Hallo, wie geht's?

Table 2: Several examples of language tag strategies.

mized by the cross-entropy loss:

$$\mathcal{L}_{\mathrm{MNMT}}(\theta) = -\sum_{j=1}^{m} \log p(y_j|\mathbf{y}^t_{<j}, \mathbf{x}^s, \mathbf{t}; \theta). \quad (2)$$

In this paper, we focus on the analysis of the placement of $\mathbf{t}$ and explore the better way to indicate the target language for zero-shot translation.

2.2 Language Tag Strategy

Pioneered by Johnson et al. (2017), the LT strategy has become the *de facto* strategy to build the unified MNMT model. Recently, many varieties are proposed to adjust the placement of the LT and we list several popular samples in Tab.2. Among these strategies, the T-Enc and S-Enc-T-Dec strategies are the most widely used ones to build MNMT models. The first proposed T-Enc strategy performs the best on zero-shot translation (Wu et al., 2021; Wicks and Duh, 2022). And S-Enc-T-Dec is also widely used in establishing MNMT models, *e.g.*, M2M100 (Fan et al., 2021), mBART (Liu et al., 2020), and so on. Although some studies (Xue et al., 2021; ElNokrashy et al., 2022) propose to place double tags on the encoder side, the capability of indicating the target language still remains insufficient (Wu et al., 2021). Therefore, this paper mainly focuses on the two most widely used strategies, *i.e,* T-Enc and S-Enc-T-Dec, to investigate the influence of the placement of LT.

2.3 Off-Target issue in Zero-Shot Translation

The *off-target* issue describes the wrong target languages translated by the MNMT models on zero-shot translation (Zhang et al., 2020). Prior studies (Gu et al., 2019; Zhang et al., 2020; Liu et al., 2021; Wang et al., 2021; Yang et al., 2021; Mao et al., 2023; Zan et al., 2023) reveal that the spurious correlations between language pairs within supervised data aggravate this issue and make efforts to overcome it in terms of adjusting training strategy, modifying the residual connection and generating auxiliary data. Besides, Wang et al. (2022c) points out that data noises also make the *off-target* issue, and Wang et al. (2022c); Jin and Xiong (2022); Chen et al. (2023) enhance the model's awareness to the vocabulary of target language during generation. However, the cause of fundamental language tag strategies, which perform various on this issue, remains unclear.

3 Probing Off-Target issue in MNMT

In this section, we probe the *off-target* issue in MNMT models with different LT strategies. Firstly, we introduce the language rate/accuracy metric on this issue (§3.1). Secondly, we conduct experiments in terms of the distribution of this issue (§3.2), the fine-grained language accuracy along decoding steps (§3.3) and the language indication in the encoder (§3.4). Lastly, we expand our conclusion to propose **LCS** to mitigate this issue (§3.5).

3.1 Metrics of Off-Target issue

To quantify this issue, we adopt the language rate as the metric to observe the error language distribution of zero-shot translation. We adopt the langdetect[3] toolkit to identify the language of generated sentences, following prior studies (Zhang et al., 2020; Wang et al., 2021; Jin and Xiong, 2022). And we calculate the language rate of language ℓ as follows:

$$Rate(\ell) = \frac{\sum_{i=1}^{N} \mathbb{1}_{lang(y^{(i)})=\ell}}{N} \times 100\%, \quad (3)$$

where $\mathrm{y}^{(i)}$ denotes the i-th generated sentence, N denotes the scale of the test set, and $lang(\cdot)$ denotes the language detect function. The **accruacy** denotes the language rate of the desired target language, where the higher language accuracy denotes the slighter *off-target* issue. Generally, language accuracy is regarded as an indicator of the performance of zero-shot translation, where the higher accuracy usually guides to the better performance.

To further observe the variation of language rate throughout decoding steps, we calculate the rate of continuous intervals with 5 words in it along generated sentences. To accurately detect the language of short text, we choose the *lingua-py*[4] toolkit, which has higher accuracy in detecting the short texts[5].

[3] https://github.com/Mimino666/langdetect

[4] https://github.com/pemistahl/lingua-py

[5] More accurate as described as their GitHub repository.

Noise?	Strategy	Acc ↑	To-Src ↓	To-En ↓	To-Other ↓
Noise	T-Enc	44.38	38.71	**3.62**	13.29
	S-Enc-T-Dec	14.09	**0.44**	72.46	13.01
	ST-Enc	2.15	3.71	81.96	**12.18**
	ST-Enc-T-Dec	**55.99**	7.22	12.08	24.71
De-Noise	T-Enc	**81.64**	2.51	**3.17**	**12.68**
	S-Enc-T-Dec	28.78	**0.38**	57.55	13.29
	ST-Enc	36.06	6.96	39.40	17.58
	ST-Enc-T-Dec	25.83	0.69	52.40	21.08

Table 3: The average language rate (%) of several LT strategies on the OPUS-100 dataset. Acc , To-Src, To-En, and To-Other denote the language rate of the expected target language, the source language, English, and other undesired languages in translation, respectively.

3.2 Error Distribution of Off-Target Issue

In this section, we first conduct experiments on the OPUS-100 dataset, which contains 100 languages, to count the error distribution of the *off-target* issue. Prior study (Wang et al., 2022c) points out that the noise in data is an important factor of the *To-Source* issue. Thus, we also count the error distribution with the denoised data[6], which filters the noise language pairs with target sentences in wrong languages, following Wang et al. (2022c).

As shown in Tab.3, the mainstream LT strategies suffer from the serious *off-target* issue. Most mainstream LT strategies endure the grave *To-English* issue, which ranges from 39.40% to 57.55% even after data denoise, signifying that the language generation is disturbed by English. Besides, the T-Enc and ST-Enc strategies tolerate the severer *To-Source* issue than other strategies, indicating that the target tag on the encoder side may be mixed with the indication to the source language. Error distributions on both data settings suggest that the placement of LT has a non-ignorable impact on the *off-target* issue. To understand the impact of LT better, we explore the language variation of language generation and the encoder's modeling tendency within different LT strategies. Specifically, we employ the widely-used T-Enc and S-Enc-T-Dec strategies on the denoised OPUS-100 dataset to conduct analysis experiments.

3.3 Fine-grained Language Accuracy along Decoding Steps

In this section, we explore the variation of language indication in generation, by calculating the fine-grained language rate throughout decoding steps.

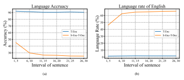


Figure 1: Fine-grained language rate of the desired target language (*i.e.*, language accuracy) and undesired English throughout decoding steps on the zero-shot testset of denoised OPUS-100, with 5 words in each interval of the final translation results.

As shown in Fig.1(a), in terms of language accuracy, the S-Enc-T-Dec strategy exhibits a considerable decrease of approximately 20%. Besides, as shown in Fig.1(b), the rate of mistranslating into English of S-Enc-T-Dec increases by around 20%. These variations signify that the indication from the S-Enc-T-Dec strategy is rapidly degraded and switched to English after generating a few tokens. Hence, we conclude that the language indication of the S-Enc-T-Dec is decreasing after a few tokens, resulting in the bias to the most common language in the training set, *i.e.*, English. Conversely, the T-Enc strategy exhibits a more sufficient and stable language indication throughout the decoding steps. Comparing both the above strategies, we conclude that the indication from the encoder could be more sufficient and stable and avoid being diminished during generation, while this setting suffers from the *To-Source* issue.

3.4 Language Indication in the Encoder

To invest the target language indication in the encoder and the *To-Source* issue, we focus on two questions: 1) *How does the encoder model the target language tag and the input sentence of the source language, to obtain sufficient target language indication?* 2) *Is there the encoder's language representation connected with the To-Source Issue?*

To answer the first question, we visualize the variation of language representation along the encoder layers, by calculating the similarity between the source and target languages in the zero-shot sentence pairs on the OPUS-100 dataset. Following Pan et al. (2021), we adopt the average-pooled encoder layer output as the sentence representation, and then calculate their cosine similarity.

As shown in Fig.2, for the 6-layer encoder, the

[6] Training details of both settings are shown in Appendix.A.

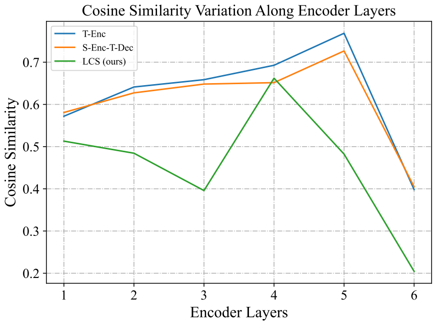


Figure 2: Curves of the similarity of the language pairs along encoder layers on the zero-shot testset of denoised OPUS-100. The higher similarity denotes the representation is more similar and language-agnostic.

Noise?	Strategy	Acc ↑	To-Src ↓	To-En ↓	To-Other ↓
Noise	T-Enc	44.38	38.71	3.62	13.29
	T-Enc-Mask	48.49	13.01	25.61	12.89
	LCS (ours)	**85.35**	**0.27**	**2.65**	**11.73**
De-Noise	T-Enc	81.64	2.51	3.17	12.68
	T-Enc-Mask	78.62	0.39	9.62	11.37
	LCS (ours)	**86.67**	**0.19**	**2.08**	**11.06**

Table 4: The average language rate (%) on the zero-shot testset of OPUS-100. T-Enc-Mask denotes the source LT is masked in each layer of 4 shallow encoder layers. Acc , To-Src, To-En, and To-Other denote the language rate of the expected target language, the source language, English, and other undesired languages in translation, respectively.

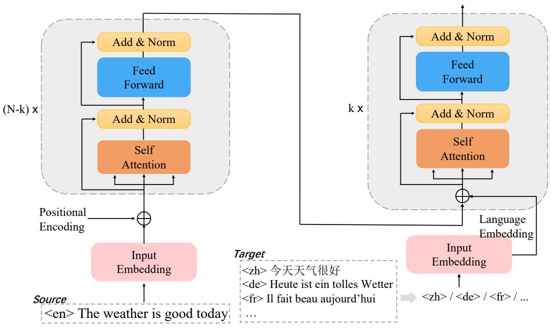


Figure 3: Illustration of the encoder of LCS. For the target, only the language tag could be seen by the encoder.

similarity scores of both LT strategies maintain the upward trend from the 1st layer to the 5th layer, and drop in the 6th layer, which is consistent with (Wu et al., 2021). The variation suggests that the encoder tends to generate language-agnostic representation from different languages first and then reduce the similarity across languages. In fact, the representation of the top layer in the T-Enc strategy tends to be target-language-specific, whereas the one of the S-Enc-T-Dec strategy tends to be source-language-specific, which is verified in Appendix D. Hence, we could answer the first question that the encoder has a first-agnositc-second-specific tendency on language representation, the target LT mainly indicates the desired target language on the top encoder layer. Further, we conjecture that the target LT is mixed with the source language features in the first stage of the tendency, resulting in the *To-Source* issue.

To answer the second question and verify our conjecture, we apply a simple operation to the T-Enc strategy. Specifically, we mask the target language tag in the 4 shallow encoder layers and restore it in the 5th layer[7]. As shown in Tab.4, this operation could significantly mitigate the *To-Source* issue in the T-Enc strategy, reducing to 13.01% (*Δ=-25.7%*) and 0.39% (*Δ=-2.12%*) on this issue. The remission on *To-Source* issue verifies our conjecture and responds to the second question that the first-agnostic-second-specific tendency introduces bias into the target LT in the first stage.

In conclusion, we summarize our findings on the impact of the placement of LT on the *off-target* issue as follows: 1) The language indication from the encoder side is more sufficient and stable, without being diminished during decoding; 2) The target language is mainly indicated on the top encoder layers, and placing the target LT at the bottom layer of the encoder introduces the *To-Source* issue.

3.5 Language Converter Strategy

Based on the conclusions in §3.3 and §3.4, we propose **L**anguage **C**onverter **S**trategy (**LCS**) to enhance the language indication to mitigate the *off-target* issue, which further improves the quality of zero-shot translation.

According to our conclusions, as shown in Fig 3, we divide the encoder layers into the shallow layers and the deep language converter layers by the k-th[8] deepest layer. In the shallow layers, we place the source language tag in front of the sentences to avoid the *To-Source* issue. In the deep language converter layers, we introduce the target language embedding as auxiliary signals to prompt the desired target language. As pointed out by prior studies (Bjerva et al., 2019; Oncevay et al., 2020; Jin and Xiong, 2022), the target language embeddings

[7] Our prior experiment shows that restoring the LT in the 5th layer performs better. And we roughly implement it via masking and adding the initial embedding of the tag.

[8] k is a hyperparameter, describing the number of the language converter layers. We set it to 2 in the 6-layer encoder and explore the selection of k in §5.2.

Model	MultiUN			TED			OPUS-100		
	Supervised	Zero-Shot	Accuracy	Supervised	Zero-Shot	Accuracy	Supervised	Zero-Shot	Accuracy
T-Enc	50.73	32.96	92.58	25.20	10.41	94.50	24.72 / 22.81	7.29 / 12.65	44.38 / 81.64
S-Enc-T-Dec	50.71	23.63	71.85	**25.27**	2.36	62.88	24.74 / 22.07	3.80 / 4.71	14.09 / 28.78
ST-Enc	50.66	1.15	0.27	24.47	1.37	13.66	24.82 / 22.82	1.92 / 7.38	2.15 / 36.06
ST-Enc-T-Dec	50.66	19.95	64.15	24.51	4.59	79.02	**25.05** / 22.11	4.23 / 3.60	55.99 / 25.83
T-Enc-T-Dec	**50.83**	32.23	91.10	24.53	12.07	95.73	24.75 / **22.89**	5.97 / 12.66	35.26 / 82.73
LCS (Ours)	50.75	**36.03***	**95.28***	**25.27**	**13.71***	**96.21***	24.80 / 22.09	**15.22*** / **15.58***	**85.35*** / **86.67***

Table 5: Experiments in several LT strategies on the MultiUN, TED, and OPUS-100 datasets. Supervised and Zero-Shot denote the average BLEU scores of the supervised and zero-shot translation directions. Accuracy denotes the averaged language accuracy (%) of zero-shot translation. "A / B" separates the scores of noise data and denoise data in OPUS-100, where ‘A’ and ‘B’ represent the result of the noise and denoised version, respectively. Results with */** are statistically better than ‘T-Enc’ in all translation directions with $p < 0.01$. **Bold** denotes the best performance.

Dataset	Langs & Dirs		Train / Valid / Test
	Supervised	Zero-Shot	
MultiUN	4 & 6	3 & 6	2M / 4K / 4K
TED	20 & 38	19 & 342	14K-22K / 5K / 5K
OPUS-100	100 & 198	6 & 30	10K-1M / 2K / 2K

Table 6: Data statistics. Langs&Dirs represents the number of languages and translation directions involved in the supervised and zero-shot translation. Train / Valid / Test represents the number of samples in each translation direction in the training / validation / test set.

contain target language-specific features, which could provide sufficient indication of the target language into the top encoder layers.

Then the final calculation of the self-attention block (Vaswani et al., 2017) in language converter layers is as follows:

$$\tilde{h}_i = h_i + e^t, \tag{4}$$

$$s = LayerNorm(\tilde{h} + SelfAttn(\tilde{h})), \tag{5}$$

where h_i denotes the i-th state of input tokens to each converter layer, s denotes the output states of self-attention block in each converter layer, $SelfAttn$ denotes the calculation of self-attention and $LayerNorm$ represents the LayerNorm function. Since the language tags have already been included in the vocabulary, LCS introduces no extra parameters. Besides, we maintain the cross-entropy loss to optimize the MNMT model.

Moreover, we place the target language tag in front of the decoder input to indicate the target language better. We verify the effectiveness of this placement in Appendix B.

4 Experiments

4.1 Experimental Setup

We conduct experiments on three popular datasets, MultiUN, TED, and OPUS-100 (Gu et al., 2019; Qi et al., 2018; Zhang et al., 2020). The statistics of the datasets of each translation direction are presented in Tab.6. (Please refer to Appendix A for more details.) For direct comparison, we report scores of noise and denoised data versions of OPUS-100. For all these datasets, English is the central language in the training sets, serving as either the source or the target language in each sentence pair. Following Johnson et al. (2017), we consider translation directions involving English as supervised translation and directions between non-English languages as zero-shot translation.

We employ open-source toolkit fairseq(Ott et al., 2019) to implement the Transformer models, with mixed precision (Ott et al., 2018). During inference, we set beam size to 5 and length penalty to 1.0, following existing studies (Wang et al., 2021; Jin and Xiong, 2022). We apply SacreBLEU to calculate BLEU and report the averaged scores for all models. More details about the dataset, training, and evaluation can be found in Appendix A.

4.2 Main Results

We respectively establish MNMT models in the several LT strategies in §2.2 and our LCS on the three datasets and list the results in Tab.5. Besides, we conduct experiments in the T-Enc-T-Dec, which seems to combine the indication from both encoder and decoder sides. Moreover, we also report the results of the denoised OPUS-100 dataset for a comprehensive comparison, where MultiUN and TED are low-noise datasets.

As shown in Tab.5, our proposed LCS yields the highest language accuracy and best performance of zero-shot translation on all datasets among all LT strategies. Specifically, compared to the widely-used T-Enc strategy, LCS effectively mitigates the *off-target* issue by improving language accuracy up to 95.28% (+*2.7%*), 96.21% (+*1.71%*), 85.35% (+*40.97%*), and 86.67% (+*5.03%*) on zero-shot translation of MultiUN, TED, OPUS-100 (noise and denoised) datasets, respectively. Furthermore, LCS outperforms the T-Enc strategy by 3.07, 3.30, 7.93, and 2.93 BLEU scores improvements on zero-shot translation of these datasets, respectively. While bringing conspicuous improvement to zero-shot translation, LCS maintains the performance of supervised translation. Unfortunately, since the language detection toolkit is lowly accurate for medium- and low-resource languages, the performance on related translation pairs is reduced in the denoised OPUS-100. In this case, LCS performs well and robust on zero-shot translation on the noise version of the OPUS-100 dataset, compared to the denoise version.

We also probe the performance of our method in prior aspects, *i.e.*, the distribution of the *off-target* issue, and the language variation in the encoder. As shown in Tab.4, LCS could yield the highest language accuracy, and the lowest rates on the *To-Source* and *To-English* issue. These scores suggest that LCS could provide the most accurate target language indication for zero-shot translation, and avoid being mixed with the indication of source language. Since LCS performs better than T-Enc on the *To-English* issue, we consider that LCS also provides sufficient and stable indication during generation. Besides, in terms of language representation variation, our proposed LCS yields the lowest score in the 6-th layer than the T-Enc strategy, meaning that LCS could generate more target-language-specific representation to indicate the target language better.

5 Analysis

In this section, we explore the details of LCS to understand it better. We invest the generalizability of LCS to other approaches (§5.1) and the effect of hyperparameter k (§5.2), and the application of LCS to deeper encoders (§5.3).

5.1 Application to Stronger Approaches

We evaluate the generalizability of LCS on OPUS-100 by applying it to the following stronger approaches:

- **Fine-Tune (FT)**. We first train the model in the S-Enc-T-Dec strategy with 100K steps, and fine-tune the model in our strategy, with the same total training steps as other models.

Model	Supervised	Zero-Shot	Accuracy
S-Enc-T-Dec	24.74 / 22.07	3.80 / 4.71	14.09 / 28.78
FT & LCS	**25.29 / 22.34**	**15.45 / 15.90**	**85.26 / 86.32**
DisPI	24.62 / **22.09**	4.80 / 4.94	18.91 / 30.36
DisPI & LCS	**24.78** / 21.87	**15.71 / 15.80**	**85.34 / 87.35**
DN†	-	-	-
DN	19.04 / **22.83**	3.02 / 12.62	21.75 / 81.66
DN & LCS	**24.32** / 22.10	**14.00 / 15.55**	**81.81 / 86.60**
LEE†	**24.98** / -	10.08 / -	79.90 / -
LEE	24.13 / 21.56	11.88 / 12.30	73.69 / 80.00
LEE & LCS	24.88 / **21.91**	**15.20 / 15.63**	**85.91 / 86.97**
CTS	**24.21 / 21.72**	12.77 / 10.34	80.09 / 62.35
CTS & LCS	24.19 / 21.52	**15.13 / 15.43**	**86.85 / 87.86**
mBART & FT	29.03 / 29.53	3.47 / 4.94	5.10 / 13.05
mBART & LCS	**30.84 / 31.33**	**21.15 / 21.24**	**86.14 / 87.51**

Table 7: Experiments about the application of LCS to other approaches. '†' represents that the results are cited from the corresponding papers, and the rest models are reproduced by us. "/" separates the scores of noise data and denoise data in OPUS-100.

- **Denosing Encoder (DN)** (Wang et al., 2021). DN introduces the denoising auto-encoder training objective to bridge the connection between zero-shot language pairs.
- **Disentangling Positional Information (DisPI)** (Liu et al., 2021). DisPI removes the residual connection of the encoder middle layer to yield the language-agnostic representation.
- **Contrastive Learning (CTS)** (Pan et al., 2021). CTS introduces the contrastive learning training objective to close the representation gap of similar sentences. We apply this objective to our shallow encoder layers since the function is similar to the first encoder stage of LCS.
- **Language Embedding Embodiment (LEE)** (Jin and Xiong, 2022). LEE adopts the target language embedding added to each state at the decoder side to indicate the desired target language without any LT strategies.
- **mBART** (Liu et al., 2020). mBART is a widely-used pretrained multilingual sequence-to-sequence model. We finetune mBART in vanilla S-Enc-T-Dec and LCS strategies, with the same training setting. Training setting of mBART is a bit different from others, details can be referred in Appendix A.

Results. As shown in Tab.7, LCS is effective in enhancing the performance of existing approaches. Specifically, with regard to zero-shot translation and language accuracy, most approaches achieve significant improvements with the application of LCS. On the OPUS-100 dataset, all approaches

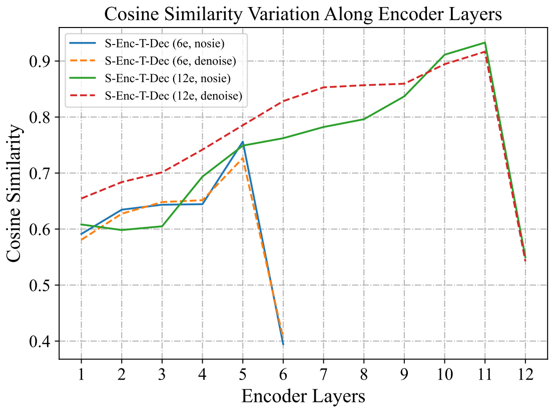


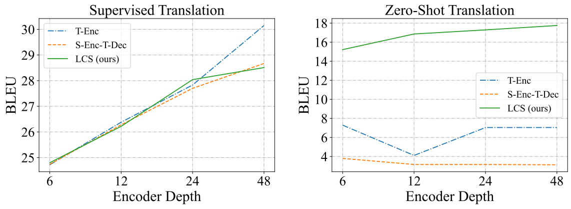
Figure 4: Curves of the similarity of the language pairs along encoder layers in S-Enc-T-Dec on the zero-shot testset of noise and denoised OPUS-100.

k	Supervised		Zero-Shot	
	6-layer	12-layer	6-layer	12-layer
2	**24.80 / 22.09**	**26.62 / 23.40**	15.22 / **15.58**	16.86 / **17.39**
3	24.64 / 21.76	26.60 / 22.85	**15.32** / 15.51	**17.01** / 16.86
4	24.13 / 21.22	2.62 / 9.03	15.06 / 15.06	2.33 / 4.29
5	23.20 / 20.81	23.75 / 19.93	14.03 / 14.75	15.40 / 14.62
6	23.62 / 20.99	22.70 / 20.15	12.79 / 14.22	14.45 / 14.66

Table 8: BLEU scores of LCS with different k on the noise and denoise OPUS-100. k denotes the hyperparameter k and 6/12-layer denotes the encoder's depth.

yield significant improvements with the application of LCS, with 11.65, 10.91, 10.98, 3.32, and 2.36 average BLEU scores improvements on the noise data version, respectively. Compared to the noise version of the OPUS-100 dataset, LCS also could yield similar improvements on the denoise version, even achieving the 15.90 BLEU score by the FT method and the 87.86% accuracy by the CTS&LCS method. Further, on the pretrained multilingual model, LCS exhibits advanced performance to enhance the performance of mBART on both supervised and zero-shot translation. While in terms of supervised translation, the application of LCS generally maintains or slightly improves the performance of most approaches. In conclusion, these results demonstrate that LCS is well-compatible with other approaches and can achieve further improvement when combined with them, without introducing extra parameters.

5.2 Selection of Hyperparameter k



(a)

(b)

Figure 5: Performance of deeper encoder on the supervised and zero-shot testset of the noise OPUS-100.

The selection of hyperparameter k is an important factor of LCS. Indeed, the selection of k mainly relies on the variation of language similarity among encoder layers, which is described in Section 3.4. We expand the variation into the deeper encoder with 12 layers, and display the variation in Fig.4. We list the performance of different values of k in the 6-layer and 12-layer encoder in Tab.8.

As the results show, the optimal selection of k is around the inflection point of the variation of language similarity. We could observe from Fig.4 and Tab.8, when the selected value k is around the inflection point, LCS could yield better translation quality. For most settings of k, the performance of LCS on zero-shot translation is much better than other methods (as shown in Tab.5 and Tab.7). We further probe the selection of k in other deeper encoders (24-layer and 48-layer), and conclude that the better range selection of k is the nearby integers of 15% of the encoder depth.

5.3 Effect of LCS on deeper encoders

We conduct experiments to compare the effectiveness of LCS with the T-Enc and S-Enc-T-Dec, where we set k to 2 for 6- and 12-layer encoders, 5 for the 24-layer encoder, and 6 for the 48-layer encoder[9]. As shown in Fig.5, although T-Enc and S-Enc-T-Dec could yield better translation on supervised translation with deeper encoders, both cannot perform much better than the 6-layer encoder on zero-shot translation. Compared to them, with deeper encoders, LCS exhibits an upward trend on zero-shot translation, where the 48-layer encoder improves around 2.50 BLEU score compared to the 6-layer encoder. Besides, in deep encoders, LCS maintains a similar supervised performance with the S-Enc-T-Dec strategy.

6 Conclusion

In this paper, we have identified that the *off-target* issue is sensitive to the placement of LT and provide the analysis for the details of the issue with

[9] In spired by Wang et al. (2022a), we train 24- and 48-layer encoder with the assistance of DeepNorm.

the prevalent LT strategies. We reveal that placing LT on the encoder side provides a more stable and sufficient language indication than the decoder side, and introducing the target information into the top layers of the encoder will mitigate the confusion between the source and the target language. Based on our findings, we propose LCS to mitigate the *off-target* issue, and further improve the performance of zero-shot translation. Extensive experiments and analysis suggest that LCS boosts the performance of zero-shot translation and significantly mitigates the *off-target* issue without introducing extra parameters. Besides, LCS could perform well in the noised data set.

Limitations

Compared to the boosted performance on zero-shot translation, LCS yields limited improvements on the supervised translation, while it is designed to enhance the language indication for zero-shot translation. We intend to explore ways to improve the performance of zero-shot translation further and achieve greater improvement in supervised translation.

Acknowledgements

The research work described in this paper has been supported by the National Nature Science Foundation of China (No. 61976016, 62376019, 61976015), and the authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Johannes Bjerva, Robert Östling, Maria Han Veiga, Jörg Tiedemann, and Isabelle Augenstein. 2019. What do language representations really represent? *Computational Linguistics*, 45(2):381–389.

Liang Chen, Shuming Ma, Dongdong Zhang, Furu Wei, and Baobao Chang. 2023. On the off-target problem of zero-shot multilingual neural machine translation. *arXiv preprint arXiv:2305.10930*.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Muhammad ElNokrashy, Amr Hendy, Mohamed Maher, Mohamed Afify, and Hany Hassan Awadalla. 2022. Language tokens: A frustratingly simple approach improves zero-shot performance of multilingual translation. *arXiv preprint arXiv:2208.05852*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Pengzhi Gao, Liwen Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2023. Improving zero-shot multilingual neural machine translation by leveraging cross-lingual consistency regularization. *arXiv preprint arXiv:2305.07310*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Renren Jin and Deyi Xiong. 2022. Informative language representation learning for massively multilingual neural machine translation. *arXiv preprint arXiv:2209.01530*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System*

Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinan Xu, Jiaan Wang, Yufeng Chen, and Jie Zhou. 2023. Unified model learning for various neural machine translation. *arXiv preprint arXiv:2305.02777*.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021. Improving zero-shot translation by disentangling positional information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1259–1273, Online. Association for Computational Linguistics.

Yijin Liu, Xianfeng Zeng, Fandong Meng, and Jie Zhou. 2023. Branchnorm: Robustly scaling extremely deep transformers. *arXiv preprint arXiv:2305.02790*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Zhuoyuan Mao, Raj Dabre, Qianying Liu, Haiyue Song, Chenhui Chu, and Sadao Kurohashi. 2023. Exploring the impact of layer normalization for zero-shot neural machine translation. *arXiv preprint arXiv:2305.09312*.

Arturo Oncevay, Barry Haddow, and Alexandra Birch. 2020. Bridging linguistic typology and multilingual machine translation with multi-view language representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2391–2406, Online. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Zhi Qu and Taro Watanabe. 2022. Adapting to non-centered languages for zero-shot multilingual translation. *arXiv preprint arXiv:2209.04138*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. 2022a. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*.

Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022b. Clidsum: A benchmark dataset for cross-lingual dialogue summarization. *arXiv preprint arXiv:2202.05599*.

Weizhi Wang, Zhirui Zhang, Yichao Du, Boxing Chen, Jun Xie, and Weihua Luo. 2021. Rethinking zero-shot neural machine translation: From a perspective of latent variables. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4321–4327, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Wang, Wenxiang Jiao, Shuo Wang, Zhaopeng Tu, and Michael R Lyu. 2022c. Understanding and mitigating the uncertainty in zero-shot translation. *arXiv preprint arXiv:2205.10068*.

Rachel Wicks and Kevin Duh. 2022. The effects of language token prefixing for multilingual machine translation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 148–153.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3001–3007, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Changtong Zan, Liang Ding, Li Shen, Yibin Lei, Yibing Zhan, Weifeng Liu, and Dacheng Tao. 2023. Unlikelihood tuning on negative samples amazingly improves zero-shot translation. *arXiv preprint arXiv:2309.16599*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Mingliang Zhang, Fandong Meng, Yunhai Tong, and Jie Zhou. 2021. Competence-based curriculum learning for multilingual machine translation. *arXiv preprint arXiv:2109.04002*.

Xiang Zhang, Ning Shi, Bradley Hauer, and Grzegorz Kondrak. 2023. Bridging the gap between BabelNet and HowNet: Unsupervised sense alignment and sememe prediction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2789–2798, Dubrovnik, Croatia. Association for Computational Linguistics.

A Dataset & Training details

We show the detail of the MultiUN[10], TED[11], and OPUS-100[12] as the following sections. We average the last five checkpoints to form the final tested checkpoint for all models in our experiments.

A.1 MultiUN

We select four languages distributed in various language families, Arabic (Ar), English (En), Russian (Ru), and Chinese(Zh), following Wang et al. (2021). In final, the training set consists of 6M sentence pairs. We select Transformer-base architecture based on post-norm to conduct experiments, and set 6 encoder/decoder layers with 8 attention heads, embedding size of 512, inner size of 2048, the dropout rate of 0.1, the maximum learning rate of 0.0007 and label smoothing rate of 0.1. We share the vocabulary for all languages and segment words into subwords using byte pair encoding (BPE) (Sennrich et al., 2016) with 40k merge operations, following Wang et al. (2021). In training, we set the maximum batch size per GPU to 4096 tokens and trained on 8 GPUs with 300K steps.

A.2 TED

It includes 60 languages in total (Qi et al., 2018) and we choose the top 20 languages following Qu and Watanabe (2022). In final, the training set consists of 3.5M sentence pairs. We choose Transformer-base architecture based on post-norm to conduct experiments, and set 6 encoder/decoder layers with 8 attention heads, embedding size of 512, inner size of 2048, dropout rate of 0.1, maximum learning rate of 0.0005 and label smoothing rate of 0.1. In this dataset, we use SentencePiece to segment words into subwords with 64k merge operations, following Qu and Watanabe (2022). In training, we set the maximum batch size per GPU to 6400 tokens and trained on 8 GPUs with 100K steps.

A.3 OPUS-100

It includes 100 languages in total and consists of 55M training sentence pairs with up to 1M samples per language pair. We choose Transformer-base architecture based on post-norm to conduct experiments, and set 6 encoder/decoder layers with 8 attention heads, embedding size of 512, inner size of

[10] https://conferences.unite.un.org/UNCORPUS

[11] https://github.com/neulab/word-embeddings-for-nmt

[12] https://opus.nlpl.eu/opus-100.php

2048, dropout rate of 0.1, maximum learning rate of 0.0005 and label smoothing rate of 0.1. In this dataset, we use SentencePiece (Kudo and Richardson, 2018) to segment words into subwords with 64k merge operations, following Jin and Xiong (2022). In training, we set the maximum batch size per GPU to 6400 and trained on 8 GPUs with 400K steps. We train models in both noise and denoise data versions with the same parameter settings and similar epochs.

A.4 mBART

Since the number of languages in mBART is less than OPUS-100, we mainly select the six languages in the zero-shot test set of OPUS-100 (*i.e.*, Arabic, German, French, Russian, and Chinese), and add English to conduct experiments. The training set is still English-centric. We select the mBART-Large[13] with 25 languages. During training, we set dropout to 0.3, and learning rate to 0.00003 with 2500 steps to warmup. We set the maximum batch size per GPU to 1024 tokens and trained on 8 GPUs with 100K steps. For fine-tuning mBART in LCS strategies, we set k to 2.

A.5 Deep Encoder Training

Inspired by the successful application of DeepNorm (Wang et al., 2022a) and BranchNorm (Liu et al., 2023), we utilize DeepNorm to stabilize the training of deep models (24-layer and 48-layer), which applies such a constraint to the early stage of model training. And the α and β are following DeepNorm, as the following formulas:

$$\begin{aligned} \alpha_{encoder} &= 0.81(N^4M)^{\frac{1}{16}}, \\ \beta_{encoder} &= 0.87(N^4M)^{-\frac{1}{16}}, \\ \alpha_{encoder} &= (3M)^{\frac{1}{4}}, \\ \beta_{decoder} &= (12M)^{-\frac{1}{4}}, \end{aligned} \tag{6}$$

where N and M denote the depth of the encoder and decoder for a standard Transformer.

A.6 Evaluation

In MultiUN, we calculate the case-insensitive sacreBLEU scores, following Wang et al. (2021), and calculate case-sensitive sacreBLEU scores for TED and OPUS-100 datasets, following Jin and Xiong (2022). We respectively average the scores of overall supervised or zero-shot translation directions to report in tables. Specifically, we apply different tokenizer for all Chinese testset[14], compared to the rest languages[15].

Dataset	Strategy	Supervised	Zero-Shot	Accuracy
MultiUN (Small)	sS-cS	**50.75**	**36.03**	**95.28**
	sS-cT	**50.75**	35.51	94.69
	sS-c_	50.73	35.93	95.12
	s_-c_	50.63	35.76	95.04
	sT-cT	34.68	11.88	18.53
	Remove	50.83	35.86	95.17
OPUS-100 (Large)	sS-cS	24.80 / 22.09	**15.22** / **15.58**	85.35 / 86.67
	sS-cT	24.82 / **22.51**	15.18 / 15.56	85.28 / 86.95
	sS-c_	**24.83** / 22.15	15.14 / 15.56	85.08 / 86.79
	s_-c_	23.84 / 21.63	14.43 / 15.10	**85.95** / **87.02**
	sT-cT	24.77 / 22.08	5.37 / 12.36	29.56 / 79.11
	Remove	24.73 / 21.96	12.22 / 13.78	52.44 / 74.95

Table 9: Experiments of the LCS strategy variants. 's/c' denotes the shallow stage or the language converter stage in the encoder of LCS, 'S/T' denotes the source or target LT in front of sentences, and '_' denotes no LT in this stage.

B Impact of the Language Tag Placement

In this section, we investigate the impact of the LT placement for LCS. We first conduct experiments to explore the influence of the source and target LT in two stages of LCS. As shown in Tab.9, when placing the source language tag on the shallow stage, the choice of language tag on the language converter stage will yield a relatively small impact on both datasets. However, placing the target LT or no placing performs unstable on the shallow stage, with bad BLEU scores. We also explore the impact of removing the target LT on the decoder side on LCS, as listed as 'Remove' in Tab.9. Compared to the vanilla LCS strategy ('sS-cS' in Tab.9), 'Remove' degrades 0.17 and 3.00 BLEU scores on zero-shot translation of two datasets, suggesting that placing the target LT on the decoder side is beneficial for LCS to perform better on zero-shot translation. Therefore, we conclude that placing the source LT on the encoder side and the target LT on the decoder side is optimal for LCS.

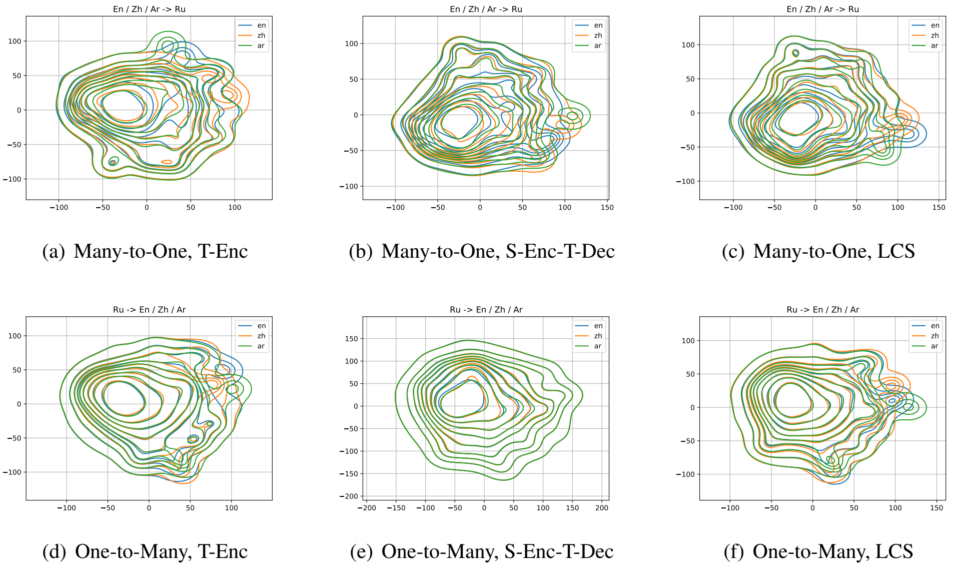
C Selection of k in deeper encoder

In this section, we supply the experiments of exploration on hyperparameter k in 24- and 48-layer

[13] `https://dl.fbaipublicfiles.com/fairseq/models/mbart/mbart.cc25.v2.tar.gz`

[14] SacreBLEU signatures: BLEU+case.mixed+numrefs.1+smooth.exp+tok.zh+version.2.0.0.

[15] SacreBLEU signatures: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.2.0.0.



(a) Many-to-One, T-Enc
(b) Many-to-One, S-Enc-T-Dec
(c) Many-to-One, LCS
(d) One-to-Many, T-Enc
(e) One-to-Many, S-Enc-T-Dec
(f) One-to-Many, LCS

Figure 6: The KDE of T-SNE reduced averaged encoder output in many-to-one and one-to-many directions.

k	Supervised		Zero-Shot	
	24-layer	**48-layer**	**24-layer**	**48-layer**
2	28.01 / 24.20	29.51 / 25.58	4.68 / 5.26	3.51 / 4.42
3	28.16 / 24.46	29.38 / **25.73**	4.55 / 5.03	4.07 / 9.48
4	**28.21** / 24.72	29.28 / 25.68	16.19 / 17.25	10.34 / 6.19
5	28.04 / **24.77**	**29.97** / 25.45	**17.29 / 18.29**	16.61 / 18.26
6	27.11 / 24.51	28.51 / 25.20	17.20 / 18.11	**17.75 / 18.65**

Table 10: BLEU scores of LCS with different k on the noise and denoise OPUS-100. k denotes the hyperparameter k and 24/48-layer denotes the encoder's depth.

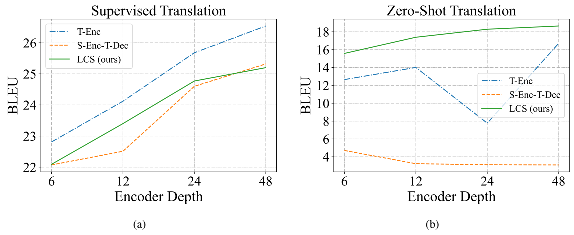


Figure 7: Translation performance of deeper encoder on the supervised and zero-shot test set of denoised OPUS-100.

encoders on the noise and denoise OPUS-100 datasets. We report the results in Tab.10.

As illustrated by Tab.10, the optimal selection of k in the 24-layer encoder is 5 and 6 in the 48-layer encoder on zero-shot translation, while keeping similar performance on supervised translation. The better range of selection of k is the nearby integers of 15% of the encoder depth.

Further, we also evaluate the performance of T-Enc, S-Enc-T-Dec, and LCS strategies on the denoised OPUS-100 dataset. Compared to the noise version, although the T-Enc could roughly yield better performance of zero-shot translation with more encoder layers, LCS could perform better than it, demonstrating that LCS is effective in boosting the performance of zero-shot translation.

D T-SNE Visualization

To detect the language-specific of the encoder output representation, we sample 2000 samples from the MultiUN test set[16], each with four languages, *i.e.*, Arabic (Ar), English (En), Russian (Ru), and Chinese(Zh). We retrieve the encoder output representation in T-Enc, S-Enc-T-Dec, and LCS strategies on many-to-one and one-to-many directions[17], and visualize them by T-SNE (Van der Maaten and Hinton, 2008), following Pan et al. (2021).

As shown in Figure 6, in the many-to-one direction (Figure 6(a), 6(b), and 6(c)), the contour lines of three strategies tend to overlap, while the contour lines of T-Enc have the highest degree to overlap with each other, and the contour lines of S-Enc-T-Dec have the lowest degree. The overlap results suggest that the language-specific of the encoder output representation is target-language-specific and the overlap ranking of the three strategy contour lines is consistent with the cosine similarity ranking in 2. However, in the one-to-many

[16] Each sentence has the corresponding translation in every language.

[17] Our models are trained in many-to-many direction

direction of S-Enc-T-Dec (Figure 6(e)), the contour lines nearly perfectly overlap with each other, suggesting that the encoder output representation of S-Enc-T-Dec tends to be source-specific rather than target-language-specific with the indication from the source language tag in deep encoder layers. In summary, the encoder output representation of the T-Enc and LCS strategy tends to be target-language-specific, while the representation of the S-Enc-T-Dec strategy tends to be source-language-specific.