switch-GLAT: MULTILINGUAL PARALLEL MACHINE TRANSLATION VIA CODE-SWITCH DECODER

Zhenqiao Song[*1], **Hao Zhou**[*1], **Lihua Qian**[1], **Jingjing Xu**[1], **Shanbo Cheng**[1], **Mingxuan Wang**[1], **Lei Li**[2]
[1]ByteDance AI Lab, Shanghai, China [2]University of California, Santa Barbara
{songzhenqiao,zhouhao.nlp,qianlihua}@bytedance.com, lilei@ucsb.edu
{chengshanbo,wangmingxuan.89}@bytedance.com, jingjingxu@pku.edu.cn

ABSTRACT

Multilingual machine translation aims to develop a single model for multiple language directions. However, existing multilingual models based on Transformer are limited in terms of both translation performance and inference speed. In this paper, we propose *switch*-GLAT, a non-autoregressive multilingual machine translation model with a code-switch decoder. It can generate contextual code-switched translations for a given source sentence, and perform code-switch back-translation, greatly boosting multilingual translation performance. In addition, its inference is highly efficient thanks to its parallel decoder. Experiments show that our proposed *switch*-GLAT outperform the multilingual Transformer with as much as 1.16 BLEU improvement and 6.6x faster decoding speed in inference.

1 INTRODUCTION

Neural machine translation (NMT) is the current state-of-the-art approach (Bahdanau et al., 2014; Sutskever et al., 2014) for machine translation in both academia (Bojar et al., 2017) and industry (Hassan et al., 2018). Recent works (Firat et al., 2016; Johnson et al., 2017; Aharoni et al., 2019; Lin et al., 2020) extend the approach to support multilingual translation, i.e. training a single model that can translate across multiple language directions. Multilingual models are appealing for several reasons. First, they can reduce the online translation service number, enabling simpler deployment (Arivazhagan et al., 2019) when plenty of translation directions are required. Additionally, multilingual training makes it possible to transfer knowledge from high-resource languages to low-resource ones, thus improving the translation quality of low-resource directions (Zoph et al., 2016; Johnson et al., 2017; Wang & Neubig, 2019).

However, most multilingual NMT systems are built upon the autoregressive architecture, which translates from left to right and thus is not efficient enough in terms of translation speed. Such efficiency problem is more serious in multilingual setting because all translation directions suffer from this slow inference speed.

A straightforward solution to improve the multilingual translation efficiency is to develop multilingual non-autoregressive translation (NAT). NAT generates translation outputs in parallel (Gu et al., 2018), which leads to significantly faster translation speed. Thanks to the recent progress of NAT (Ghazvininejad et al., 2019; Gu et al., 2019; Deng & Rush, 2020), current state-of-the-art NAT models have achieved comparable BLEU scores (Li et al., 2018; Wei et al., 2019; Qian et al., 2020) with their auto-regressive counterparts. Among them, the glancing transformer (GLAT) proposed by Qian et al. (2020) is a representative work, which even outperforms many strong autoregressive translation systems in BLEU score on German-English translation task of WMT21 (Qian et al., 2021).

In this paper, we argue that multilingual NAT models is not only superior in efficiency, but also can achieve better multilingual translation accuracy, due to its capability of generating high quality code-switched translations (Lin et al., 2020; Yang et al., 2020; Jose et al., 2020). In particular, we propose *switch*-GLAT, a carefully designed multilingual version of GLAT, which can outperform multilingual Transformer in both speed and translation quality. Generally, the main idea of *switch*-GLAT is to

[*]Corresponding author.

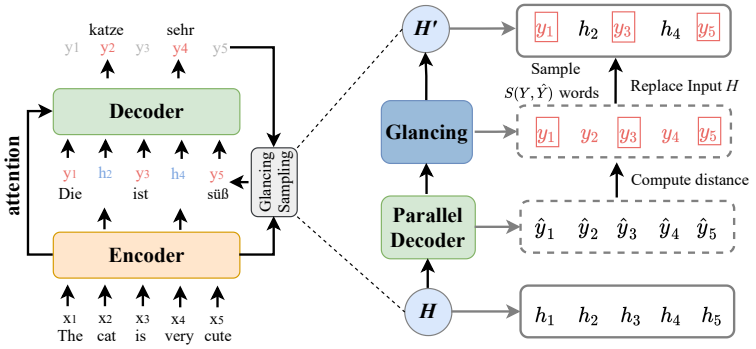


Figure 1: The overall architecture of GLAT. The left half shows the training process of GLAT while the right half details the glancing sampling strategy.

employ a *code-switch decoder*, which can generate *contextual code-switched* translations (instead of using dictionary for replacement) for a given source sentence, and then perform *code-switch back-translation* to boost the multilingual translation performance. In more details, the glancing sampling module in GLAT enables *switch*-GLAT to generate partial translation based on others. Then combined with its multilingual and non-autoregressive characteristics, *switch*-GLAT could output code-switched translations with the employment of *token-level language tags* instead of sentence-level ones in auto-regressive models. Ultimately, *switch*-GLAT can generate contextual translated words at arbitrary positions of the target sentence in arbitrary languages. This greatly improves the multilingual translation performance when we reverse the pairs of source to contextual code-switched target sentences, for training in a back-translation fashion (so called code-switch back-translation).

We conduct extensive experiments on 3 merged translation datasets: WMT with four language pairs (both close languages and distant ones) and WMT with 10 language pairs. *switch*-GLAT shows consistent improvements over autoregressive multilingual baselines on all datasets, validating that *switch*-GLAT can achieve better multilingual translation performance simultaneously with a faster decoding speed. We further evaluated the cross-lingual representations through word induction and sentence retrieval tasks. The results demonstrated the proposed code-switch back-translation benefits better-aligned cross-lingual representations.

2 Background

Multilingual Neural Machine Translation (MNMT) Given a source sentence $X = \{x_1, x_2, ..., x_M\}$ with length M and its target sentence $Y = \{y_1, y_2, ..., y_N\}$ with length N, MNMT leverages the standard bilingual neural machine translation models and extends the source and target inputs respectively with a source and target language token src and tgt. This results in $X' = \{\mathrm{src}, x_1, x_2, ..., x_M\}$ and $Y' = \{\mathrm{tgt}, y_1, y_2, ..., y_N\}$.

MNMT is generally modeled from X' to Y' with Transformer (Vaswani et al., 2017). Transformer consists of stacked encoder and decoder layers, which are jointly trained to maximize the conditional probability of Y' given X':

$$P(Y'|X') = \sum_{i=1}^{N} \log P(y_i|y_{<i}, X', \mathrm{tgt}; \theta) \tag{1}$$

where θ are the trainable model parameters.

Glancing Transformer prposed by Qian et al. (2020) is a NAT architecture which achieves top results in machine translation with 8x $\sim$ 15x speedup. It performs two-pass decoding in training but is still fully non-autoregressive in inference. In the first decoding pass, given the encoder F_e and decoder F_d, $H_d^0 = \{h_1^0, h_2^0, ..., h_N^0\}$ is the decoder input either gathered from the encoder output using *soft copy* (Wei et al., 2019) or *full mask* (Ghazvininejad et al., 2019), and then Y is predicted as:

$$\hat{Y} = F_d(H_d^0, F_e(X; \theta); \theta) \tag{2}$$

where θ are the trainable model parameters. Then glancing transformer (GLAT) adopts glancing sampling strategy to sample a subset of Y according to its distance with $\hat{Y}$, thus resulting in $GS(Y, \hat{Y})$.

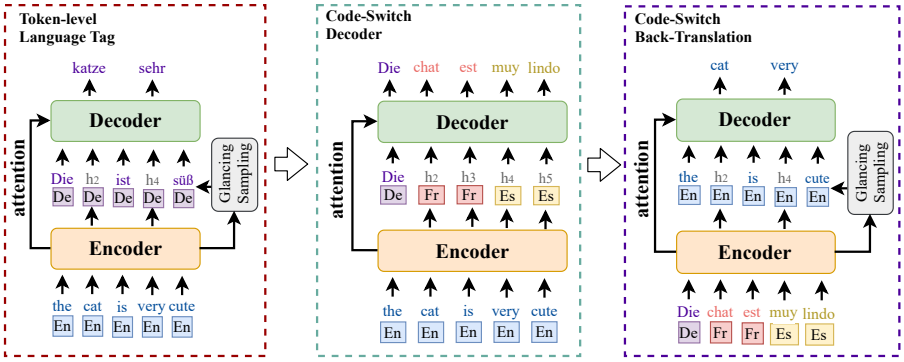


Figure 2: The overall architecture of *switch*-GLAT. The left module shows that GLAT is extended to a multilingual version with the token-level language tag, the middle one describes how the code-switched sentence is produced through the code-switch decoder, and the right module introduces the code-switch back-translation process, in which the generated code-switched target and its paired source is swapped to enhance training.

Finally, GLAT predicts the target sequence Y based on this subset and source sentence X in the second decoding pass as follows:

$$L_{\text{GLAT}} = -\sum_{y_t \in \overline{\mathbb{GS}(Y,\hat{Y})}} \log P(y_t|\mathbb{GS}(Y,\hat{Y}), X; \theta) \tag{3}$$

where H_d^0 is updated accordingly and $\overline{\mathbb{GS}(Y,\hat{Y})}$ denotes the remaining subset of Y after removing the sampled tokens. The overall architecture of GLAT is shown in Figure 1.

The twice decoding in training makes GLAT capable of predicting some partial translations based on others. Then combined with multilingual and non-autoregressive characteristics, GLAT has the potential to generate contextual code-switched outputs. In the next section, we will show how to modify the GLAT architecture to achieve this.

3 Proposed Method: *switch*-GLAT

This section will detail the proposed *switch*-GLAT. The training objective can be factorized into two parts. One aims to make the model have multilingual translation ability, and the other to achieve better-aligned cross-lingual representations to boost translation performance, respectively denoted as L_{multi} and L_{csbt}. The overall objective can be formulated as follows:

$$L = L_{\text{multi}} + \lambda \cdot L_{\text{csbt}} \tag{4}$$

where λ plays the role of a "temperature" to schedule the importance of code-switch back-translation. As the training progresses, λ is gradually increased, which allows for more complex data instances to be involved, i.e. code-switched translations incorporating more languages and higher switching ratios. This process encourages model to align similar words from different languages into the same vector space, thus boosting machine translation performance. The overall architecture is shown in Figure 2.

3.1 Code-Switch Decoder

switch-GLAT leverages the general parallel decoder of GLAT and extends it into a code-switch decoder by employing the token-level language tag. The code-switch decoder first gains multilingual translation ability through multilingual training, and then it can generate code-switched translations in arbitrary languages with the help of token-level language tag.

Specifically, given a multilingual corpora $\mathbb{D} = \{\mathbb{D}^l\}_{l=1}^{L}$ consisting of L language pairs, the loss L_{multi} is then defined as:

$$\begin{aligned} L_{\text{multi}} &= \sum_{\mathbb{D}^l \in \mathbb{D}} \sum_{(X_j^l, Y_j^l) \in \mathbb{D}^l} \{L_{\text{tag}}(Y_j^l|X_j^l;\theta_M) + L_{\text{len}}^l(j)\} \\ L_{\text{len}}^l(j) &= -P(L_j^l) \log \hat{P}(L_j^l|[F_e(X_j^l;\theta_M); E_{\text{src}}; E_{\text{tgt}}];\theta_M) \end{aligned} \tag{5}$$

where $\mathbb{D}^l = \{(X_j^l, Y_j^l)\}_{j=1}^{N_l}$ is a parallel corpus with size N_l and $L_{\text{len}}^l(j)$ is the length prediction loss of the j-th pair. $P(L_j^l)$ is the real length distribution of target sentence, and $\hat{P}(L_j^l)$ is the predicted one based on the concatenation of encoder output as well as source and target language embeddings. θ_M are trainable model parameters. L_{tag} is the accordingly updated GLAT training loss (Equation 3) incorporating the source and target language tags:

$$\begin{aligned} L_{\text{tag}}(Y_j^l|X_j^l;\theta_M) &= -\sum_{y_t \in \overline{\mathbb{GS}(Y_j^l,\hat{Y}_j^l)}} \log P(y_t|\mathbb{GS}(Y_j^l,\hat{Y}_j^l), X_j^l, \text{src}, \text{tgt}; \theta_M) \\ \hat{Y}_j^l &= F_d(\tilde{H}_d^0, F_e(X_j^l, \text{src}; \theta), \text{tgt}; \theta_M) \end{aligned} \tag{6}$$

To involve the indicative language tag, we add it to the first layer input and final layer output at each position of both encoder and decoder as follows:

$$\begin{aligned} \tilde{f}_i^0 &= f_i^0 + E_{\text{src}}; \tilde{f}_i^K = f_i^K + E_{\text{src}} \\ \tilde{h}_j^0 &= h_j^0 + E_{\text{tgt}}; \tilde{h}_j^K = h_j^K + E_{\text{tgt}} \end{aligned} \tag{7}$$

where f_i^0 denotes the first encoder layer input at position i and h_j^0 denotes the first decoder layer input at position j. Correspondingly, f_i^K and h_j^K denote the last layer output. src and tgt are respectively the source and target individual tags, while E_{src} and E_{tgt} are their corresponding representations. The overall prediction process of pair (X_j^l, Y_j^l) is illustrated in the left module of Figure 2.

Through the multilingual training process, *switch*-GLAT can translate between different languages using the indicative language tags, of which the decoder is called code-switch decoder. It has the ability to generate contextual translated words in arbitrary languages due to its token-level characteristics.

3.2 Code-Switch Back-Translation

Thanks to the code-switch decoder, we can perform code-switch back-translation (CSBT), which is critical because it encourages model to align the produced words and the original ones into the same vector space according to their similar context information. Better-aligned cross-lingual representations benefit better translation performance.

Specifically, a subset $\mathbb{D}_S = \{(X_i, Y_i, l_i^{\text{src}}, l_i^{\text{tgt}})\}_{i=1}^S$ of size S is first sampled, where l_i^{src} and l_i^{tgt} are respectively source and target languages of i-th pair. Then, Y_i is masked with a given rate P_M, leading to $\tilde{Y}_i$. Subsequently, the masked positions of $\tilde{Y}_i$ can be decoded into a third randomly sampled language by leveraging the token-level language tag. Thus, the final decoded sequence $\hat{Y}_i$ consists of contextual tokens from mixed languages, which will in turn be taken as the source side input and the original source sentence as the target side input. This process results in a code-switch back-translation corpus $\mathbb{D}_C = \{(\hat{Y}_i, X_i)\}_{i=1}^S$, which is illustrated in the middle module of Figure 2.

The dynamically generated code-switch translations can augment data instance distribution to enhance model training as illustrated in the right module of Figure 2. Loss L_{csbt} is defined as:

$$L_{\text{csbt}} = \sum_{(\hat{Y}_i, X_i) \in \mathbb{D}_C} L_{\text{tag}}(X_i|\hat{Y}_i;\theta_M) \tag{8}$$

As the training continues, the masked rate P_M and number of mixed languages are gradually increased. Specifically, the value of P_M is iterated from 0.1 to 0.5 with step size 0.1 every 10 epochs. In the first iteration of P_M, the number of mixed languages is set to 1. Afterwards, it will be increased to one-third of the total. Through this process, abundant code-switched sentences can be generated, which helps to learn better-aligned cross-lingual representations.

3.3 Scheduled Code-Switch Training

The influence of multilingual training and code-switch back-translation is balanced by λ. As the training progresses, λ is increased and more complicated data instances are involved, which means that code-switch back-translation becomes more important. We use the step function to evolve the value of λ:

$$\lambda(t) = f(t) + \lambda(0), \quad f(t) = \begin{cases} 0, & t < E, \\ 1, & t \geq E \end{cases} \tag{9}$$

where $\lambda(t)$ is the importance value at step t and E is a pre-defined changing point to incorporate code-switched translations. $\lambda(0)$ is the starting value and we set it to 0. As the training goes on, the code-switch back-translation and model training are iterated, leading to well-aligned cross-lingual representations and improved machine translation performance.

4 EXPERIMENTS

4.1 DATASETS

To better measure the effects of multilingualism at different levels, we test our proposed *switch*-GLAT on the following three merged datasets: (1) WMT-EDF: We collect 4 language pairs from WMT-14 English (En) $\leftrightarrow$ German (De) and English (En) $\leftrightarrow$ French (Fr). All three languages belong to Indo-European language family and are relatively *close* on linguistics. (2) WMT-EFZ: We also collect 4 language pairs from WMT-14 English (En) $\leftrightarrow$ French (Fr) and WMT-17 English (En) $\leftrightarrow$ Chinese (Zh), which are *distant* languages on linguistics and their relationships are more difficult to learn. (3) WMT-many: We also gather 10 language pairs from WMT-14 English (En) $\leftrightarrow$ German (De), English (En) $\leftrightarrow$ French (Fr), WMT-16 English (En) $\leftrightarrow$ Russian (Ru), English (En) $\leftrightarrow$ Romanian (Ro) and WMT-17 English (En) $\leftrightarrow$ Chinese (Zh) to test *switch*-GLAT on more *diverse* language pairs.

4.2 BASELINE MODELS

We compare *switch*-GLAT against the following representative **multilingual baselines**: (1) **M-Transformer**: We set a multilingual transformer baseline, which also uses token-level language tags on both source and target sides. (2) **CLSR**: Zhang et al. (2021) present CLSR that learns to route between language-specific or shared pathways from the data itself to obtain the sharing structure. (3) **Adapter**: Bapna et al. (2019) propose to inject a lightweight adapter layer for each language in MNMT to extract some language-specific features. We reimplement Adapter since its code is not available currently. (4) **MNAT**: We also set a multilingual NAT baseline (Gu et al., 2018) modeled by adding token-level language tags to the vanilla NAT. All model settings are the same as *switch*-GLAT.

Besides, we also report the performance of two typical **bilingual models**: (1) **Transformer**: We use the code released by Vaswani et al. (2017) to implement Transformer. (2) **GLAT**: GLAT is the backbone of our *switch*-GLAT and it achieves comparable results to Transformer.

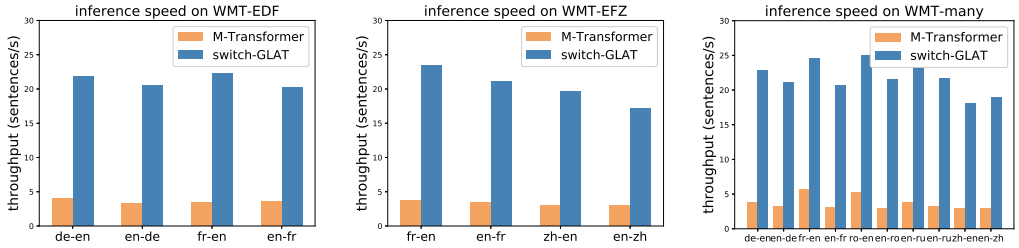
To better analyze the influence of different components in our *switch*-GLAT, we also conduct **ablation tests** as follows: (1) ***switch*-GLAT-w/o-glancing**: *switch*-GLAT removes glancing sampling strategy in training. (2) ***switch*-GLAT-w/o-CSBT**: *switch*-GLAT is trained without code-switch back-translation, which means the model is only trained on the golden parallel corpora. (3) ***switch*-GLAT-w/o-CCS**: *switch*-GLAT constructs the code-switched data by simply replacing aligned words using a bilingual dictionary instead of contextualized code-switching.

Models	WMT-EDF						WMT-EFZ				
	En-De	De-En	En-Fr	Fr-En	Avg	Speed	En-Fr	Fr-En	En-Zh	Zh-En	Avg
Bilingual models											
Transformer	27.77	31.55	38.80	37.35	**33.86**	1.4×	38.80	37.35	23.60	24.05	**30.95**
GLAT	26.09	30.53	38.62	34.44	32.42	6.1×	38.62	34.44	21.05	22.89	29.25
Multilingual models											
M-Transformer	26.21	31.61	38.45	35.77	33.01	1.0×	38.45	35.84	20.75	22.49	29.38
CLSR	23.83	31.83	39.10	35.29	32.48	0.9×	38.06	36.12	20.31	21.51	29.00
Adapter	22.40	29.94	41.18	36.11	32.41	0.9×	40.69	35.64	19.61	21.84	29.45
MNAT	14.27	22.73	24.99	26.12	22.02	6.4×	20.18	20.94	7.67	7.64	14.11
switch-GLAT	25.28	31.51	41.00	36.12	**33.47**	6.6×	41.79	37.33	19.35	23.69	**30.54**
– w/o glancing	18.20	24.13	26.67	30.23	24.81	6.3×	21.87	22.89	7.99	8.48	15.31
– w/o CSBT	24.71	29.84	37.14	34.61	31.75	6.4×	35.78	34.22	18.37	21.01	27.34
– w/o CCS	25.00	30.42	37.89	34.99	32.07	6.3×	36.66	34.46	18.47	21.56	27.79

Table 1: Translation performance (BLEU) on WMT-EDF/EFZ. Avg means the average BLEU score. The models above dash line are autoregressive ones, while those below are non-autoregressive ones.

Models	WMT-many										
	En-De	De-En	En-Fr	Fr-En	En-Ro	Ro-En	En-Ru	Ru-En	En-Zh	Zh-En	Avg
Bilingual models											
Transformer	27.77	31.55	38.80	37.35	33.01	33.59	28.22	29.89	23.60	24.05	**30.78**
GLAT	26.09	30.53	38.62	34.44	31.83	32.59	25.42	28.13	21.05	22.89	29.20
Multilingual models											
M-Transformer	23.22	29.67	35.51	34.11	34.40	35.82	24.68	29.25	16.59	20.61	28.39
CLSR	21.89	30.23	34.67	33.38	30.95	35.80	19.05	30.96	16.32	20.60	27.39
Adapter	23.59	30.01	38.75	40.20	30.91	32.32	22.91	26.68	18.16	20.97	28.45
MNAT	8.52	14.59	12.68	20.11	13.12	21.77	7.67	16.00	4.72	6.04	12.52
switch-GLAT	24.22	31.04	39.11	36.10	32.18	32.85	24.03	28.21	16.70	21.52	**28.60**
– w/o glancing	7.29	16.91	11.54	21.16	14.03	23.95	7.62	18.86	7.07	6.89	13.53
– w/o CSBT	22.89	28.78	32.22	34.31	30.31	33.06	22.46	27.13	15.27	19.06	26.54
– w/o CCS	22.69	28.91	34.73	34.23	30.63	33.50	21.69	26.97	14.93	19.84	26.81

Table 2: Translation performance (BLEU) on WMT-many. Avg means the average score. The models above dash line are autoregressive ones, while those below dash line are non-autoregressive ones.



(a) WMT-EDF (b) WMT-EFZ (c) WMT-many

Figure 3: Inference speed evaluated on throughputs.

4.3 Training and Inference Details

We use 6 layers for encoder and parallel decoder. The model hidden size d_{model} and feed-forward hidden size d_{ff} are set to 512 and 2048 respectively. The number of attention head is set to 8. All the datasets are tokenized and segmented into subword units using BPE encodings (Sennrich et al., 2016). The vocabulary size is set to 85k for WMT-EDF and 95k for WMT-EFZ/many.

Following the previous works (Gu et al., 2018; Li et al., 2020; Ma et al., 2019), the sequence-level knowledge distillation is used for all datasets, and the Transformer-base architecture (Vaswani et al., 2017) is employed as the teacher model.

The changing point E is set to $300,000$ steps and sampling number S is set to $300,000$ for each pair. The mini-batch size is set to 64k tokens and the maximum training step is $1,200,000$. The model is trained with 8 NVIDIA Tesla V100 GPU cards. We follow the default parameters of Adam optimizer (Kingma & Ba, 2014) and learning rate schedule in Vaswani et al. (2017). Dropout annealing strategy (Rennie et al., 2015) is applied to stable training and the initialized dropout rate is set to 0.3. In training, data from different language pairs are sampled according to a multinomial distribution rebalanced by a temperature of 0.3 (Conneau et al., 2019).

In inference, following Qian et al. (2020), *switch*-GLAT performs one-iteration parallel decoding. Moreover, we also leverage the common practice of noise parallel decoding (Gu et al., 2018; Ma et al., 2019) and the number of length reranking candidate is set to 5.

4.4 Main Results

4.4.1 Results on WMT-EDF/EFZ

The left and right halves of Table 1 respectively show the results of WMT-EDF and WMT-EFZ.

***switch*-GLAT outperforms all multilingual competitors on the average score in both settings.** Specifically, in WMT-EDF scenario, our model achieves the best average BLEU score of 33.47 among all multilingual competitors, especially exceeding multilingual Transformer with a fairly significant margin of 0.46 point. Moreover, it performs better than CLSR and Adapter with an improvements of 0.99 and 1.06 points respectively. The same phenomena can be observed in WMT-EFZ scenario where the language pairs are more distant on linguistics. Our interpretation is that *switch*-GLAT has the ability to produce context-dependent code-switched translations with the code-switch decoder. The augmented code-switched translations can help model align the generated tokens and the original ones into the same vector space due to their similar context information, thus boosting machine translation performance (Mikolov et al., 2013).

***switch*-GLAT can largely speed up decoding.** In WMT-EDF scenario, the decoding speed tested on Fr-En pair is reported in Table 1, which is evaluated in the throughput of each model (full speed values are illustrated in Figure 3 (a) and Figure 3 (b)). Table 1 shows that the decoding speed of *switch*-GLAT is 6.6 times faster than multilingual transformer. Additionally, Figure 3 (a) and (b) also illustrate that the throughput of our model is significantly larger than multilingual transformer on all directions. These validate that the *switch*-GLAT can greatly improve decoding efficiency thanks to its parallel decoder. Note that the length predictor is jointly trained incorporating source and target language embeddings, thus leading to different length preference in different models.

***switch*-GLAT performs better than GLAT on average.** In both scenarios, *switch*-GLAT achieves higher accuracy in 3 out of 4 directions than GLAT, especially X-to-En directions. These indicate that multilingual training makes it possible to transfer knowledge so that performance of X-to-En direction is enhanced due to the augmented data in English on the target side.

4.4.2 Results on WMT-many

Table 2 shows the results on the gathered WMT dataset from 10 language pairs.

***switch*-GLAT performs better than multilingual baselines in most cases.** It is primarily because *switch*-GLAT is able to produce diverse contextual code-switched translations, which encourage the model to automatically align the generated word vectors and the original ones due to their similar context information. Well-aligned representations are beneficial for improving machine translation performance. Notably, Adapter performs much worse than multilingual Transformer on En-Ro and En-Ru directions as the original paper interprets that Adapter do not perform well in low-resource settings (the reverse directions are enhanced by the English data from other language pairs).

***switch*-GLAT performs slightly worse than GLAT.** In this scenario, the failure of *switch*-GLAT may be owing to the limitation of model capacity. However, *switch*-GLAT outperforms GLAT when the model translates from a foreign language to English. One interpretation is that all the parallel corpora including English on either source or target side can lead to a better parameter estimation of the English decoder (Firat et al., 2016).

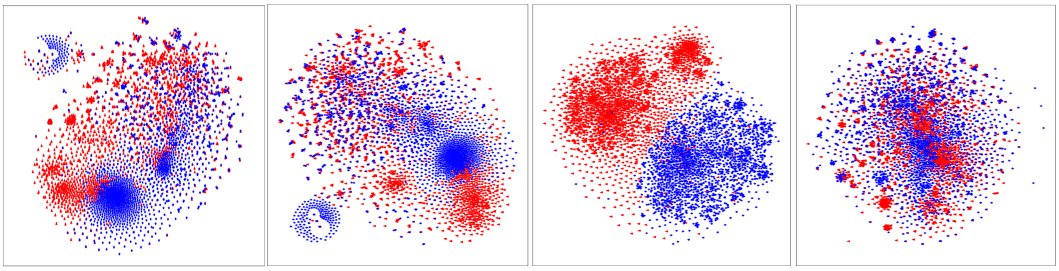
***switch*-GLAT can enlarge the power of efficient decoding.** The decoding speed in this setting are illustrated in Figure 3 (c), showing that the throughput of our model is much larger than multilingual transformer on 10 directions. It demonstrates that *switch*-GLAT can enlarge the power of decoding efficiency since all translation directions can be accelerated within a single model.

4.5 Ablation study

The last three lines in Table 1 and Table 2 show the results of ablation study.

Glancing sampling strategy plays an important role in *switch*-GLAT. After removing the glancing sampling strategy, the performance drops the most. It validates that glancing sampling is critical in training an advanced parallel multilingual translation model, which makes it suitable to generate high-quality code-switched translations.

Code-switch back-translation promotes multilingual translation. Excluding code-switch back-translation or employing code-switched data constructed by dictionary substitution both harms translation performance, of which the former hurts more. It validates that applying code-switched data can significantly increase multilingual translation performance, especially those generated by incorporating context information instead of dictionary replacing.



(a) Transformer (b) GLAT (c) M-Transformer (d) *switch*-GLAT

Figure 4: Representations learned by (a) Transformer (Vaswani et al., 2017), (b) GLAT (Qian et al., 2020), (c) M-Transformer and (d) *switch*-GLAT, projected to 2D.

Models	Word Induction					Sentence Retrieval				
	En-De	De-En	En-Fr	Fr-En	Avg	En-De	De-En	En-Fr	Fr-En	Avg
M-Transformer	30.5	31.4	33.9	37.6	33.4	19.0	22.7	21.1	24.0	21.7
CLSR	25.3	25.7	36.3	36.1	30.8	19.3	18.7	25.6	28.5	23.0
Adapter	32.5	34.3	39.7	41.3	36.9	33.5	29.8	27.5	35.0	31.4
MNAT	24.4	24.1	33.2	33.9	28.9	14.7	18.5	17.3	16.5	16.7
switch-GLAT	34.5	35.8	41.5	45.4	**39.3**	34.5	37.7	38.2	36.8	**36.8**
– w/o glancing	23.7	23.8	31.2	31.9	27.6	17.7	22.8	18.3	19.1	19.4
– w/o CSBT	28.2	29.3	32.3	36.1	31.5	18.3	21.4	19.3	20.8	19.9
– w/o CCS	28.4	29.3	32.4	36.3	31.6	17.4	22.9	20.0	20.2	20.1

Table 3: Results of quality analyses. Avg means the average accuracy.

5 Analysis

In this section, we will make a rigorous analysis to reveal the reasons why our *switch*-GLAT can achieve better translation performance than multilingual transformer. All tasks are tested in WMT-EDF scenario.

5.1 Visualization

There is much smaller gap in the cross-lingual representations learned by our *switch*-GLAT. To intuitively illustrate how well the cross-lingual representations are aligned, we plot a two-dimensional projection of representations learned by four models in Figure 4. Here multilingual models are trained on the merged WMT-EDF dataset, while bilingual ones on the English $\rightarrow$ German part. We use the t-SNE algorithm (van der Maaten Laurens & Hinton, 2008) to perform the projection. English and German words are gold word pairs from Open Multilingual WordNet (OMW) datasets (Bond & Foster, 2013). English words are displayed in blue color and German words in red. We find there is much more overlapping between blue and red areas in Figure 4 (d), validating that *switch*-GLAT can produce better-aligned cross-lingual representations than other models.

5.2 Cross-Lingual Word Induction

The cross-lingual representations are aligned well at word level. We assess the cross-lingual word induction performance to see how well the similar words from different languages are close to each other in the learned vector space (Gouws et al., 2015). Specifically, golden word pairs are extracted from the OMW datasets (Bond & Foster, 2013), resulting in $10,585$ En-De and $12,511$ En-Fr pairs. Top-1 accuracy is shown in the left half of Table 3. It shows that *switch*-GLAT significantly outperforms all autoregressive multilingual competitors on average, indicating that cross-lingual representations in our model are aligned well at word level.

5.3 PARALLEL SENTENCE RETRIEVAL

The cross-lingual representations are aligned well at sentence level. To further assess the quality of the cross-lingual representations at sentence level, we employ parallel sentence retrieval (PSR) task. PSR aims to extract parallel sentences from a comparable corpus between two languages. Following the settings of XTREME (Hu et al., 2020), we use the Tatoeba dataset (Artetxe & Schwenk, 2019). Cosine similarity is leveraged to search the nearest neighbour and the search accuracy of each model is reported in the right half of Table 3. It can be seen that *switch*-GLAT achieves the highest accuracy among all competitors, validating that better contextualized embeddings are obtained in *switch*-GLAT as the sentence embeddings are calculated with regarding all tokens in it.

6 RELATED WORK

6.1 MULTILINGUAL NEURAL MACHINE TRANSLATION

Early works extend the NMT model proposed by Bahdanau et al. (2014) to a multilingual version based either on LSTM (Hochreiter & Schmidhuber, 1997) or GRU (Cho et al., 2014). Dong et al. (2015) propose a one-to-many translation model (from English into 4 languages) by adding a dedicated decoder per target language, showing improvements over strong single-pair baseline. Zoph & Knight (2016) present a multi-source NMT which significantly outperforms a single-source model. Firat et al. (2016) propose a many-to-many model involving 10 language pairs by leveraging separate encoder and decoder per language as well as a sharing attention mechanism. Further, Ha et al. (2016) and Johnson et al. (2017) propose to train a multilingual model with a shared encoder-decoder-attention network across all languages to perform many-to-many translation. Recently, Transformer (Vaswani et al., 2017) is applied as the backbone of MNMT by many works. The Multi-Distillation (Tan et al., 2019) is trained to simultaneously fit the training data and match the outputs of individual bilingual models through knowledge distillation. Bapna et al. (2019) present to inject a language-specific adaptation layer into a pretrained model to keep the language-specific features. Aharoni et al. (2019) propose a multilingual NMT model involving up to 102 languages which achieves good zero-shot performance. All the above MNMT systems are based on autoregressive architectures which follow a sequential generation process and are limited to slow decoding speed.

6.2 NON-AUTOREGRESSIVE NEURAL MACHINE TRANSLATION

Recently, NAT has attracted much attention due to its ability of parallel decoding. A fully non-autoregressive model is first proposed by Gu et al. (2018), based on which latent variables are introduced to deal with dependencies among target words (Ma et al., 2019; Bao et al., 2019; Ran et al., 2021). Other branches of fully non-autoregressive systems either try to transfer knowledge from autoregressive models (Li et al., 2018; Wei et al., 2019; Guo et al., 2020; Sun & Yang, 2020), or apply different training objectives (Libovický & Helcl, 2018; Shao et al., 2020; Ghazvininejad et al., 2020; Qian et al., 2020). Besides, some works build NAT systems with structured prediction to model inter-dependencies among output words. Sun et al. (2019) propose to incorporate a structured inference module into the non-autoregressive models. Deng & Rush (2020) present a Markov transformer to perform cascaded decoding. Additionally, serveral semi-autoregressive models are also constructed to refine the outputs with multi-pass iterative decoding (Lee et al., 2018; Gu et al., 2019; Ghazvininejad et al., 2019; Kasai et al., 2020). All these works are limited to bilingual translation tasks. To our best knowledge, *switch*-GLAT is the first one to perform multilingual neural machine translation based on a non-autoregressive architecture.

7 CONCLUSION

This paper proposes *switch*-GLAT, a non-autoregressive multilingual neural machine translation model. It incorporates a switch-decoder, which can produce contextualized code-switched sentences and perform code-switch back-translation. Through this process, the multilingual translation performance and cross-lingual representations can both be improved. Besides, its parallel decoder enables a highly efficient inference. Experiments on three datasets suggest that *switch*-GLAT is superior to multilingual transformer in both effectiveness and efficiency. Further analyses also demonstrate the effectiveness of the learned contextual cross-lingual representations.

REFERENCES

Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3874–3884, 2019.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Yu Bao, Hao Zhou, Jiangtao Feng, Mingxuan Wang, Shujian Huang, Jiajun Chen, and Lei Li. Non-autoregressive transformer by position learning. *arXiv preprint arXiv:1911.10677*, 2019.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*, 2019.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pp. 169–214, 2017.

Francis Bond and Ryan Foster. Linking and extending an open multilingual wordnet. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, pp. 1352–1362, 2013.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

Yuntian Deng and Alexander Rush. Cascaded text generation with markov transformers. *Advances in Neural Information Processing Systems*, 33, 2020.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 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)*, pp. 1723–1732, 2015.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 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*, pp. 866–875, 2016.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, 2019.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. Aligned cross entropy for non-autoregressive machine translation. In *International Conference on Machine Learning*, pp. 3515–3523. PMLR, 2020.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 748–756, 2015.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018.

Jiatao Gu, Changhan Wang, and Jake Zhao. Levenshtein transformer. *arXiv preprint arXiv:1905.11006*, 2019.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7839–7846, 2020.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*, 2016.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pp. 4411–4421. PMLR, 2020.

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

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. A survey of current datasets for code-switching research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 136–141. IEEE, 2020.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. Non-autoregressive machine translation with disentangled context transformer. In *International Conference on Machine Learning*, pp. 5144–5155. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*, 2018.

Xiaoya Li, Yuxian Meng, Arianna Yuan, Fei Wu, and Jiwei Li. Lava nat: A non-autoregressive translation model with look-around decoding and vocabulary attention. *arXiv preprint arXiv:2002.03084*, 2020.

Zhuohan Li, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. Hint-based training for non-autoregressive translation. In *EMNLP-IJCNLP*, 2018.

Jindřich Libovickỳ and Jindřich Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3016–3021, 2018.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2649–2663, 2020.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. Flowseq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4273–4283, 2019.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. Glancing transformer for non-autoregressive neural machine translation. *arXiv preprint arXiv:2008.07905*, 2020.

Lihua Qian, Yi Zhou, Zaixiang Zheng, Yaoming Zhu, Zehui Lin, Jiangtao Feng, Shanbo Cheng, Lei Li, Mingxuan Wang, and Hao Zhou. The volctrans glat system: Non-autoregressive translation meets wmt21. *arXiv preprint arXiv:2109.11247*, 2021.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. Guiding non-autoregressive neural machine translation decoding with reordering information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13727–13735, 2021.

Steven J Rennie, Pierre L Dognin, Xiaodong Cui, and Vaibhava Goel. Annealed dropout trained maxout networks for improved lvcsr. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5181–5185. IEEE, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pp. 1715–1725, 2016.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 198–205, 2020.

Zhiqing Sun and Yiming Yang. An em approach to non-autoregressive conditional sequence generation. In *International Conference on Machine Learning*, pp. 9249–9258. PMLR, 2020.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng. Fast structured decoding for sequence models. *Advances in Neural Information Processing Systems*, 32:3016–3026, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. *arXiv e-prints*, pp. arXiv–1902, 2019.

van der Maaten Laurens and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

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

Xinyi Wang and Graham Neubig. Target conditioned sampling: Optimizing data selection for multilingual neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5823–5828, 2019.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1304–1312, 2019.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. Csp: Code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2624–2636, 2020.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*, 2021.

Barret Zoph and Kevin Knight. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 30–34, 2016.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1568–1575, 2016.

A APPENDIX

A.1 COMPARING WITH VANILLA BACK-TRANSLATION

We conduct ablation study of using vanilla back-translation (*switch*-GLAT-with-vanilla BT) in Table 4. It shows that though the model using vanilla back-translation outperforms the one not using any augmentation strategy (*switch*-GLAT-w/o-CSBT), it can be further improved by leveraging code-switch back-translation, validating that our proposed code-switch back-translation is favorable.

Models	WMT-EFZ				
	En-Fr	Fr-En	En-Zh	Zh-En	Avg
switch-GLAT	41.79	37.33	19.35	23.69	**30.54**
– with vanilla BT	39.39	36.63	19.21	22.37	29.30
– w/o CSBT	35.78	34.22	18.37	21.01	27.34

Table 4: Ablation study of using vanilla back-translation on WMT-EFZ.

A.2 APPLYING CSBT TO AUTOREGRESSIVE MODEL

We also apply our proposed code-switch back-translation (CSBT) to multilingual transformer, and the results are shown in Table 5. It shows that CSBT also boosts the performance of multilingual Transformer. Though M-Transformer-with-CSBT performs slightly better than our switch-GLAT, our model could significantly improve inference efficiency.

Models	WMT-many										
	En-De	De-En	En-Fr	Fr-En	En-Ro	Ro-En	En-Ru	Ru-En	En-Zh	Zh-En	Avg
M-Transformer	23.22	29.67	35.51	34.11	34.40	35.82	24.68	29.25	16.59	20.61	28.39
– with CSBT	24.52	29.96	37.34	34.93	34.65	35.72	25.02	29.17	17.82	20.30	**28.94**
switch-GLAT	24.22	31.04	39.11	36.10	32.18	32.85	24.03	28.21	16.70	21.52	28.60
– w/o CSBT	22.89	28.78	32.22	34.31	30.31	33.06	22.46	27.13	15.27	19.06	26.54

Table 5: Translation performance of applying code-switch back-translation to multilingual Transformer.

A.3 TRANSFER ABILITY

To validate that our model can improve the performance of low-resource languages, we train our *switch*-GLAT on the subset of TED dataset that is merged from English (En)-Spanish (Es), English-French (Fr) and English-Portuguese (Pt) pairs. En-Es and En-Fr pairs have about four times as much data as En-Pt pair. The results are shown in Table 6. It can be seen that our switch-GLAT can significantly improve the performance of low-resource languages with as much as 20.14 and 1.18 BLEU points respectively compared with GLAT and Transformer.

To further validate the zero-shot ability of our model, we evaluated it (trained on WMT-many) on the test set of WMT 2019 German-French. Table 7 shows that the zero-shot ability of switch-GLAT and multilingual Transformer are both poor, but applying self pivot to these two models can achieve good performance.

Models	TED						
	En-Es	Es-En	En-Fr	Fr-En	En-Pt	Pt-En	Avg
Transformer	34.62	38.52	37.78	36.34	27.79	30.94	**34.33**
GLAT	34.87	37.89	36.64	36.58	10.08	11.98	28.00
switch-GLAT-w/o-CSBT	34.61	38.55	35.83	37.28	21.61	31.55	33.23
switch-GLAT	35.02	39.01	36.58	38.12	23.78	**32.12**	34.10

Table 6: Translation performance of TED subsets.

Models		WMT-many		
		German-French	French-German	Avg
M-Transformer	zero-shot	2.88	3.00	2.94
	self pivot	30.74	22.27	**26.48**
switch-GLAT	zero-shot	2.09	2.60	2.34
	self pivot	28.03	19.85	23.94

Table 7: Zero-shot performance of our *switch*-GLAT. Self pivot on German-French denotes first German to English and then English to French. So does French-German.

A.4 Results with a Big Model

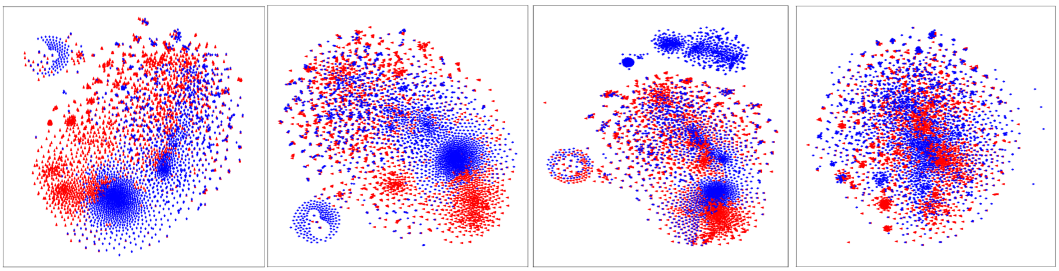
As capacity is crucial for multilingual models, we conduct experiments to show how *switch*-GLAT performance compares as model size grows. We adopt a standard Transformer-large architecture (Vaswani et al., 2017), which employs 6-layer encoder, 6-layer decoder and 1024 dimension on 16 heads. The results are shown in Table 8. It shows that in a multi-direction scenario, GLAT performs better than *switch*-GLAT when trained on a base model, but increasing model size brings opposite results, demonstrating that model capacity is quite critical for multilingual models.

A.5 Appended Visualization

We also visualize the representations learned by CLSR since it is also a representative multilingual baseline. Again, we find there is much more overlapping between blue and red areas in Figure 5 (d), validating that *switch*-GLAT can produce better-aligned cross-lingual representations than other models.

A.6 Hyper-parameter Selection

We tune the training step of changing lambda (100000 to max steps with step size 100000), sample number S (100000 to max data size with step size 200000) and initialized dropout rate (0.1 to 0.5



(a) Transformer (b) GLAT (c) CLSR (d) *switch*-GLAT

Figure 5: Representations learned by (a) Transformer, (b) GLAT, (c) CLSR and (d) *switch*-GLAT, projected to 2D.

Models	WMT-many										
	En-De	De-En	En-Fr	Fr-En	En-Ro	Ro-En	En-Ru	Ru-En	En-Zh	Zh-En	Avg
Base models											
GLAT	26.09	30.53	38.62	34.44	31.83	32.59	25.42	28.13	21.05	22.89	29.20
switch-GLAT	24.22	31.04	39.11	36.10	32.18	32.85	24.03	28.21	16.70	21.52	28.60
Large models											
GLAT	26.55	31.18	39.15	35.53	31.70	31.55	27.01	29.40	19.50	21.61	29.32
switch-GLAT	25.83	31.59	42.62	36.17	33.56	34.57	25.24	29.89	18.55	21.12	**29.91**

Table 8: Results of different model size on WMT-many.

with step size 0.1)with grid search on the validation set of WMT-EDF, and found that the values described in the paper performed best.

P_M	*switch*-GLAT	*switch*-GLAT-w/o-CCS
0.0	The longest journey begins with the first step.	The longest journey begins with the first step.
0.1	The longest voyage begins with the first step.	The longest journey begins with the first étape.
0.3	Le plus long viaje begins with the first step.	The longest journey empezar with the premier pisar.
0.5	Die längste Reise empieza con the first step.	The höchste viajar empezar with das first step.

Table 9: Some examples of generated code-switched sentences respectively by our switch-GLAT and dictionary replacement. French, Spanish and German are represented in red, purple and yellow respectively.

A.7 Case Study

To gain an insight on how well our model can generate code-switched sentences, we provide some examples in Table 9. It shows that constructing code-switched sentences by dictionary replacement may cause errors. For example, *pisar* is a verb while the original word *step* is a noun, and using a verb here is not appropriate.