Improving Multilingual Translation by Representation and Gradient Regularization

Yilin Yang\textsuperscript{1,*}, Akiko Eriguchi\textsuperscript{2}, Alexandre Muzio\textsuperscript{2}, Prasad Tadepalli\textsuperscript{1}, Stefan Lee\textsuperscript{1}, Hany Hassan\textsuperscript{2}
\textsuperscript{1}Oregon State University  
\textsuperscript{2}Microsoft

\textsuperscript{1}\{yangyil,tadepall,leestef\}@oregonstate.edu  
\textsuperscript{2}\{akikoe,alferre,hanyh\}@microsoft.com

Abstract

Multilingual Neural Machine Translation (NMT) enables one model to serve all translation directions, including ones that are unseen during training, i.e. zero-shot translation. Despite being theoretically attractive, current models often produce low quality translations – commonly failing to even produce outputs in the right target language. In this work, we observe that off-target translation is dominant even in strong multilingual systems, trained on massive multilingual corpora. To address this issue, we propose a joint approach to regularize NMT models at both representation-level and gradient-level. At the representation level, we leverage an auxiliary target language prediction task to regularize decoder outputs to retain information about the target language. At the gradient level, we leverage a small amount of direct data (in thousands of sentence pairs) to regularize model gradients. Our results demonstrate that our approach is highly effective in both reducing off-target translation occurrences and improving zero-shot translation performance by +5.59 and +10.38 BLEU on WMT and OPUS datasets respectively. Moreover, experiments show that our method also works well when the small amount of direct data is not available.\textsuperscript{1}

1 Introduction

With Neural Machine Translation becoming the state-of-the-art approach in bilingual machine translations (Bahdanau et al., 2015; Vaswani et al., 2017), Multilingual NMT systems have increasingly gained attention due to their deployment efficiency. One conceptually attractive advantage of Multilingual NMT (Johnson et al., 2017) is its capability to translate between multiple source and target languages with only one model, where many directions\textsuperscript{2} are trained in a zero-shot manner.

Despite its theoretical benefits, Multilingual NMT often suffers from target language interference (Johnson et al., 2017; Wang et al., 2020c). Specifically, Johnson et al. (2017) found that Multilingual NMT often improves performance compared to bilingual models in many-to-one setting (translating other languages into English), yet often hurts performance in one-to-many setting (translating English into other languages). Several other works (Wang et al., 2018; Arivazhagan et al., 2019; Tang et al., 2020) also confirm one-to-many translation to be more challenging than many-to-one.

Another widely observed phenomenon is that the current multilingual system on zero-shot translations faces serious off-target translation issue (Gu et al., 2019; Zhang et al., 2020) where the generated target text is not in the intended language. For example, Table 1 shows the percentage of off-target translations appearing between high-resource languages. These issues exemplify the internal failure of multilingual systems to model different target languages. This paper focuses on reducing off-target translation, since it has the potential to improve the quality of zero-shot translation as well as general translation accuracy.

Previous work on reducing the off-target issue often resorts to back-translation (BT) techniques (Sennrich et al., 2015). Gu et al. (2019)\textsuperscript{*} Work done while interning at Microsoft.

\textsuperscript{1}Codes and rebuilt OPUS data at: https://github.com/yilinyang7/fairseq_multi_fix

\textsuperscript{2}Most NMT dataset are English-centric, meaning that all training pairs include English either as source or target.

|       | WMT Baseline | Fr-De | De-Fr | Cs-De | De-Cs |
|-------|--------------|-------|-------|-------|-------|
| OPUS  | Fr-De        | 95.15%| 93.70%| 68.85%| 91.20%|
| Baseline | De-Fr | 51.60%| 39.80%| 13.10%| 20.50%|

Table 1: Off-target translation percentages on WMT and OPUS Testsets.
employs a pretrained NMT model to generate BT parallel data for all \( O(N^2) \) English-free\(^3\) directions and trains the multilingual systems on both real and synthetic data. Zhang et al. (2020) instead fine-tune the pretrained multilingual system on BT data that are randomly generated online for all zero-shot directions. However, leveraging BT data for zero-shot directions has some weaknesses:

- The need for BT data grows quadratically with the number of languages involved, requiring significant time and computing resources to generate the synthetic data.
- Training the multilingual systems on noisy BT data would usually hurt the English-centric performance (Zhang et al., 2020).

In this work, we propose a joint representation-level and gradient-level regularization to directly address multilingual system’s limitation of modeling different target languages. At representation-level, we regulate the NMT decoder states by adding an auxiliary Target Language Prediction (TLP) loss, such that decoder outputs are retained with target language information. At gradient-level, we leverage a small amount of direct data (in thousands of sentence pairs) to project the model gradients for each target language (TGP for Target-Gradient-Projection). We evaluate our methods on two large scale datasets, one concatenated from previous WMT competitions with 10 languages, and the OPUS-100 from Zhang et al. (2020) with 95 languages. Our results demonstrate the effectiveness of our approaches in all language pairs, with an average +5.59 and +10.38 BLEU gain across zero-shot pairs, and 24.5% → 0.9% and 65.8% → 4.7% reduction to off-target rates on WMT-10 and OPUS-100 respectively. Moreover, we show the off-target translation not only appears in the zero-shot directions, but also exists in the English-centric pairs.

2 Approach

In this section, we will illustrate the baseline multilingual models and our proposed joint representation and gradient regularizations.

2.1 Baseline Multilingual NMT Model

Following Johnson et al. (2017), we concatenate all bilingual parallel corpora together to form the training set of a multilingual system, with an artificial token appended to each source sequence to specify the target language.

Specifically, given a source sentence \( x^i = (x^i_1, x^i_2, ..., x^i_{|x^i|}) \) in language \( i \) and the parallel target sentence \( y^j = (y^j_1, y^j_2, ..., y^j_{|y^j|}) \) in language \( j \), the multilingual model is trained with the following cross-entropy loss:

\[
L_{NMT} = - \sum_{t=1}^{|y^j|} \log P_\theta(y^j_t | x^i_1, (j), y^j_{1..(t-1)}),
\]

(1)

where \((j)\) is the artificial token specifying the desired target language, and \(P_\theta\) is parameterized using an encoder-decoder architecture based on a state-of-the-art Transformer backbone (Vaswani et al., 2017).

We then train the multilingual system on the concatenated parallel corpus of all available language pairs in both forward and backward directions, which is also referred to as a many-to-many multilingual system. To balance the training batches between high-resource and low-resource language pairs, we adopt a temperature-based sampling to up/down-sample bilingual data accordingly (Arivazhagan et al., 2019). We set the temperature \( \tau = 5 \) for all of our experiments.

2.2 Representation-Level Regularization: Target Language Prediction (TLP)

As shown in Table 1, the current multilingual baseline faces serious off-target translation issue across the zero-shot directions. With the multilingual decoder generating tokens in a wrong language, its decoder states for different target languages are also mixed and not well separated, in spite of the input token \((j)\). We thus introduce a representation-level regularization by adding an auxiliary Target Language Prediction (TLP) task to the standard NMT training.

Specifically, given the source sentence \( x = (x_1, x_2, ..., x_{|x|}) \) and a desired target language \((j)\), the model generates a sequence of decoder states \( z = (z_1, z_2, ..., z_{|y|}) \). As the system feeds \( z \) through a classifier and predicts tokens (in Equation 1), we feed \( z \) through a LangID model to classify the desired target language \((j)\). TLP is then optimized with the cross-entropy loss:

\[
L_{TLP} = - \log M_\theta(z, (j)),
\]

(2)

\(^3\)We denote translation directions that do not involve English as English-free directions.

\(^4\)\(z\) is of the same length as the system translation \(\hat{y}\).
where the LangID model $M_\theta$ is parameterized as a 2-layer Transformer encoder with a LangID classifier on the top. TLP loss is then linearly combined with $L_{NMT}$ with a coefficient $\alpha$. Empirically, we found that any value from $\{0.1, 0.2, 0.3\}$ for $\alpha$ performs similarly well.

$$L = (1 - \alpha) \cdot L_{NMT} + \alpha \cdot L_{TLP}. \quad (3)$$

**Implementation** We implement the LangID model as a 2-layer Transformer encoder with input from the multilingual decoder states to classify the target language. We add to the decoder states a sinusoidal positional embedding for position information. We implement two common approaches to do classification: **CLS_Token** and **Meanpooling**. For **CLS_Token**, we employ a BERT-like (Devlin et al., 2018) CLS token and feed its topmost states to the classifier. For **Meanpooling**, we simply take the mean of all output states and feed it to the classifier. Their comparison is shown in Section 4.1.

### 2.3 Gradient-Level Regularization: Target-Gradient-Projection (TGP)

Although the TLP loss helps build more separable decoder states, it lacks reference signals to directly guide the system on how to model different target languages. Inspired by recent gradient-alignment-based methods (Wang et al., 2020a; Yu et al., 2020; Wang et al., 2020d, 2021), we propose **Target-Gradient-Projection (TGP)** to guide the model training with constructed oracle data, where we project the training gradient to not conflict with the oracle gradient.

**Creation of oracle data** Similar to Wang et al. (2020a, 2021), we build the oracle data from multilingual dev set, since the dev set is often available and is of a higher quality than the training set. More importantly, for some zero-shot pairs, we are able to include hundreds or thousands of parallel samples from the dev set. We construct the oracle data by concatenating all available dev sets and grouping them by the target language. For example, the oracle data for French would include every other language to French. The detailed construction of oracle data is specific to each dataset, and described in Section 3.4. The dev set often serves to select the best checkpoint for training, thus we split the dev set as 80% for oracle data and 20% for checkpoint selection.

**Implementation** Contrary to standard multilingual training, where a training batch consists of parallel data from different language pairs, we group the training data by the target language after the temperature-based sampling (Section 2.1). By doing so, we treat the multilingual system as a multi-task learner, and translations into different languages are regarded as different tasks. Similarly, we construct the oracle data individually for each target language, whose gradients would serve as guidance to the training gradients.

For each step, we obtain the training gradients $g_{train}^i$ for target language $i$, and the gradients of the corresponding oracle data $g_{oracle}^i$. Whenever we observe a conflict between $g_{train}^i$ and $g_{oracle}^i$, which is defined as a negative cosine similarity, we project $g_{train}^i$ into the normal plane of $g_{oracle}^i$ to de-conflict (Yu et al., 2020).

$$g_{train}^i = g_{train}^i - \frac{g_{train}^i \cdot g_{oracle}^i}{\|g_{oracle}^i\|^2} \cdot g_{oracle}^i \cdot (4)$$

The detailed algorithm is illustrated in Algorithm 1.

```plaintext
Algorithm 1: Target-Gradient-Projection

Input: Involved language set $L$; Pre-trained model $\theta$; Training data $D_{train}$; Oracle data $D_{oracle}$; Update frequency $n$

1. Initialize step $t = 0$, $\theta_0 = \theta$
2. while not converged do
   \begin{align*}
   & \triangleright Update oracle data gradients \\
   & \text{if } t \pmod{n} = 0 \text{ then} \\
   & \quad \text{for } i \text{ in } L \text{ do} \\
   & \quad \quad g_{oracle}^i = \nabla_{\theta_i} L(\theta_i, B^i) \\
   & \quad \end{align*}
3. end
4. Sample minibatches grouped by target language $B = \{T_i\}$
5. for $i$ in $L$ do
6. \begin{align*}
   & g_{train}^i = \nabla_{\theta_i} L(\theta_i, T^i) \\
   & \text{if } g_{oracle}^i \cdot g_{train}^i < 0 \text{ then} \\
   & \quad g_{train}^i = g_{train}^i - \frac{g_{train}^i \cdot g_{oracle}^i}{\|g_{oracle}^i\|^2} \cdot g_{oracle}^i \\
   & \end{align*}
7. end
8. Update $t \leftarrow t + 1$
9. Update $\theta_i$ with gradient $\sum_i g_{train}^i$
```

The detailed algorithm is illustrated in Algorithm 1.
## 3 Experimental Setup

### 3.1 Datasets: WMT-10

Following Wang et al. (2020b), we collect parallel data from publicly available WMT campaigns\(^5\) to form an English-centric multilingual WMT-10 dataset, including English and 10 other languages: French (Fr), Czech (Cs), German (De), Finnish (Fi), Latvian (Lv), Estonian (Et), Romanian (Ro), Hindi (Hi), Turkish (Tr) and Gujarati (Gu). The size of bilingual datasets range from 0.08M to 10M, with five language pairs above 1M (Fr, Cs, De, Fi, Lv) and five language pairs below 1M (Et, Ro, Hi, Tr, Gu). We use the same dev and test set as Wang et al. (2020b). Since the WMT data does not include zero-shot dev or test set, we have created 1k multi-way aligned dev and test sets for all involved languages based on the WMT2019 test set. For evaluation, we picked 6 language pairs (12 translation directions) to examine zero-shot performance, including pairs of both high-resource languages (Fr-De and De-Cs), pairs of high- and low-resource languages (Ro-De and Et-Fr), and pairs of both low-resource languages (Et-Ro and Gu-Tr). The detailed dataset statistics can be found in Section A.1.

### 3.2 Datasets: OPUS-100

To evaluate our approaches in the massive multilingual settings, we adopt the OPUS-100 corpus from Zhang et al. (2020)\(^6\). OPUS-100 is also an English-centric dataset consisting of parallel data between English and 100 other languages. We removed 5 languages (An, Dz, Hy, Mn, Yo) from OPUS, since they are not paired with a dev or test set. However, while constructing the oracle data from its multilingual dev set, we found that the dev and test sets of OPUS-100 are noticeably noisy since they are directly sampled from web-crawled OPUS collections\(^7\). As shown in Table 2, several dev sets have significant overlaps with their test sets. 15.26% of dev set samples appear in the test set on average across all language pairs. This is a significant flaw of the OPUS-100 (v1.0) that previous works have not noticed. To fix this, we rebuild the OPUS dataset as follows. Without significantly modifying the dataset, we add an additional step of de-duplicating both the training and dev sets against the test\(^8\), and moving data from training set to complement the dev set due to de-duplication. We additionally sampled 2k zero-shot dev set using OPUS sampling scripts\(^9\) to match the released 2k zero-shot test set. The detailed dataset statistics can be found in section\(^10\).

### 3.3 Training and Evaluation

For both WMT-10 and OPUS-100, we tokenize the dataset with the SentencePiece model (Kudo and Richardson, 2018) and form a shared vocabulary of \(64k\) tokens. We employ the Transformer-Big setting (Vaswani et al., 2017) in all our experiments on the open-sourced Fairseq Implementation\(^11\) (Ott et al., 2019). The model is optimized using Adam (Kingma and Ba, 2014) with a learning rate of \(5 \times 10^{-4}\) and 4000 warm-up steps. The multilingual model is trained on 8 V100 GPUs with a batch size of 4096 tokens and a gradient accumulation of 16 steps, which effectively simulates the training on 128 V100 GPUs. Our baseline model is trained with 50k steps, while it usually converges

| Code | #Test | #Dev | Overlap(%) |
|------|-------|------|------------|
| Tk   | 1852  | 1852 | 97.46%     |
| Ig   | 1843  | 1843 | 96.31%     |
| Li   | 2000  | 2000 | 87.75%     |
| Yi   | 2000  | 2000 | 83.95%     |
| Zu   | 2000  | 2000 | 83.45%     |

Table 2: The top-5 most overlapped dev and test set on OPUS-100, the overlapping rate is calculated as the percentage of dev set that appears in the test. The average overlapping rate between dev and test is 15.26% across all language pairs.

---

\(^5\)http://www.statmt.org/wmt19/translation-task.html  
\(^6\)https://opus.nlpl.eu/opus-100.php  
\(^7\)https://opus.nlpl.eu/  
\(^8\)We keep the OPUS-100 test set as is to make a fair comparison against previous works, although there are also noticeable duplicates within the test set.  
\(^9\)https://github.com/EdinburghNLP/opus-100-corpus  
\(^10\)Our rebuilt OPUS dataset is released at https://github.com/yilinyang7/fairseq_multi_fix  
\(^11\)https://github.com/pytorch/fairseq
Table 3: Comparing TLP approaches on WMT-10. BLEU is averaged across all English-centric directions.

| TLP       | $\alpha$ | Avg. BLEU |
|-----------|----------|-----------|
| Baseline  | 0        | 23.58     |
| Meanpooling | 0.1     | 23.85     |
|           | 0.2      | 23.73     |
|           | 0.3      | 23.93     |
| CLS_Token | 0.1      | 23.81     |
|           | 0.2      | 23.76     |
|           | 0.3      | Diverged  |

Table 4: Comparing TGP granularity on WMT-10. BLEU is averaged across all English-centric directions.

| TGP       | Avg. BLEU |
|-----------|-----------|
| Baseline  | 23.58     |
| Model-wise | 24.35   |
| Layer-wise | 23.77   |
| Matrix-wise | 23.90 |

much earlier. For evaluation, we employ beam search decoding with a beam size of 5 and a length penalty of 1.0. The BLEU score is measured by the de-tokenized case-sensitive SacreBLEU\textsuperscript{12} (Post, 2018).

In order to evaluate the off-target translations, we utilize off-the-shelf LangID model from FastText (Joulin et al., 2016) to detect the languages of translation outputs.

3.4 Construction of Oracle Data

On WMT-10, we use our human labelled multi-way dev set together with the original English-centric WMT dev set to construct the oracle data. On OPUS-100, we similarly combine the zero-shot dev set with original OPUS dev set for oracle data. On OPUS, we further merge oracle data that consists of only English-centric dev sets, since it empirically obtains similar performance while exhibiting noticeable speedups. The statistics of the constructed oracle data is shown in Section A.3.

4 Results

In this section, we will demonstrate the effectiveness of our approach on both WMT-10 and OPUS-100 datasets. The full results are documented separately in Tables 5, 6, and 7 for WMT-10, and Tables 8 and 9 for OPUS-100.

4.1 TLP Results

Hyper-parameter Tuning Table 3 shows the comparison between TLP implementations on the WMT-10 dev set. We observe that the Meanpooling approach for TLP is both more stable and delivers slightly better performance. In all the following experiments, we use the Meanpooling approach for TLP, with $\alpha = 0.3$ on WMT-10 and $\alpha = 0.1$ on OPUS-100.

Performance From Tables 5 and 6 (row 4 vs. row 2), we can see that TLP outperforms baselines in most En-X and X-En directions and all English-free directions as shown in Table 7 (row 4 vs. row 2). On average, TLP gains +0.4 BLEU on En-X, +0.28 BLEU on X-En and +2.12 BLEU on English-free directions. TLP also significantly reduces the off-target rate from 24.5\% down to 6.0\% (in Table 7). Meanwhile on OPUS-100, TLP performs similarly in English-centric directions (in Table 8) while yielding +0.77 BLEU improvement on English-free directions, together with a 65.8\% → 60.5\% drop in off-target occurrences (in Table 9).

These results demonstrate that by adding an auxiliary TLP loss, multilingual models much better retain information about the target language, and moderately improved on English-free pairs.

4.2 TGP Results

Settings Similar to Yu et al. (2020); Wang et al. (2020d), the conflict detection and de-conflict projection of TGP training could be done with different granularities. We compare three options: (1) model-wise: flatten all parameters into one vector, and perform projection on the entire model; (2) layer-wise: perform individually for each layer of encoder and decoder; (3) matrix-wise: perform individually for each parameter matrix. From Table 4, we found operating on the model-level gives...
Table 5: BLEU scores of English → 10 languages translation on WMT-10. ✓ * denotes TGP training in a zero-shot manner for all evaluated English-free pairs. “Off-Tgts” column reports the average off-target rates from FastText LangID model, while the off-target rate on the references is 0.81%.

| X → En | Bilingual | Baseline | + Finetune | Ours (Baseline + ___) |
|--------|-----------|----------|------------|----------------------|
|        | ✓         | ✓ ✓      | ✓ ✓        | ✓ ✓                  |
| Avg    | 26.22     | 26.30    | 27.60      | 27.78                |
| Off-Tgts | 0.23%   | 0.13%    | 0.20%      | 0.19%                |

Table 6: BLEU scores of 10 languages → English translation on WMT-10. ✓ * denotes TGP training in a zero-shot manner for all evaluated English-free pairs. “Off-Tgts” column reports the average off-target rates from FastText LangID model, while the off-target rate on the references is 0.12%.

| X → En | Bilingual | Baseline | + Finetune | Ours (Baseline + ___) |
|--------|-----------|----------|------------|----------------------|
|        | ✓         | ✓ ✓      | ✓ ✓        | ✓ ✓                  |
| Avg    | 22.96     | 22.62    | 26.50      | 27.80                |
| Off-Tgts | 0.30%   | 0.23%    | 0.20%      | 0.19%                |

The best performance, and as a result all our TGP experiments are done on the model-level. We then perform TGP training for 10k steps on the 40k-step pretrained model.

**Performance** In Tables 5, 6 and 7 (row 5 vs. 2), TGP gains significant improvements on all directions of WMT-10: averaging +1.23 BLEU on En-X, +1.38 BLEU on X-En and +5.57 BLEU on English-free directions, while also reducing the off-target rates from 24.5% down to only 0.9%. Similar gains could also be found on OPUS-100 (in Tables 8 and 9): +3.65 BLEU on En-X, +1.32 BLEU on X-En and +10.63 BLEU on English-free, and a whopping 65.8% → 4.8% reduction to off-target occurrences. These results demonstrate the overwhelming effectiveness of TGP on all translation tasks as well as on reducing off-target cases.

**Learning curves** Figures 1 and 2 illustrate the learning curves of TGP on WMT-10. Different from the baseline curves, TGP observes a slight increase in training loss, which shows TGP as a regularizer prevents the model from overfitting to the training set. Meanwhile, a steady increase on both English-centric and English-free test BLEU.
Table 7: BLEU scores of English-free translations on WMT-10. ✓ denotes TGP training in a zero-shot manner for all evaluated English-free pairs. As a reference, the average off-target rate reported by FastText LangID model is 0.68% on the references.

| En-Free | TLP | TGP | Fr-De | De-Cs | Ro-De | Et-Fr | Et-Ro | Gu-Tr | BLEU | Off-Tgt | Avg | Avg (%) |
|---------|-----|-----|-------|-------|-------|-------|-------|-------|------|--------|-----|---------|
| 1 | Pivoting | - | - | 24.9 | 19.3 | 19.4 | 18.9 | 19.1 | 18.8 | 16.2 | 20.9 | 16.4 | 16.8 | 5.2 | 6.4 | 16.86 | 1.1% |
| 2 | Baseline | - | - | 18.5 | 12.8 | 15.8 | 13.6 | 17.5 | 16.0 | 10.3 | 13.7 | 12.5 | 14.4 | 0.9 | 1.9 | 12.33 | 24.5% |
| 3 | + Finetune | ✓ | - | 24.2 | 18.4 | 18.8 | 18.2 | 21.8 | 18.2 | 16.6 | 20.9 | 16.1 | 19.5 | 5.6 | 6.9 | 17.13 | 2.0% |
| 4 | Ours (Baseline + _) | ✓ | ✓ | 25.2 | 20.9 | 21.7 | 19.7 | 23.0 | 19.8 | 16.1 | 21.4 | 16.7 | 20.8 | 6.4 | 7.3 | 17.92 | 0.9% |
| 5 | ✓ ✓ | 24.2 | 18.3 | 20.0 | 18.3 | 22.3 | 19.5 | 21.5 | 16.1 | 20.1 | 5.6 | 7.1 | | | | |

Table 8: Average test BLEU for High/Medium/Low-resource language pairs on OPUS-100 dataset. All denotes the average BLEU for all language pairs. ✓ ✓ denotes TGP training in a zero-shot manner for all evaluated English-free pairs.

| English-Centric | TLP | TGP | X → English | English → X |
|-----------------|-----|-----|-------------|-------------|
|                 | High | Med | Low | All | High | Med | Low | All |
| 1 | Zhang et al. (2020) (24L) | - | - | 30.29 | 32.58 | 31.90 | 31.36 |
| 2 | Baseline | - | - | 30.27 | 33.50 | 31.94 | 31.61 |
| 3 | + Finetune | ✓ | - | 19.85 | 29.93 | 36.49 | 26.57 |
| 4 | Ours (Baseline + _) | ✓ | ✓ | 29.96 | 38.32 | 41.94 | 35.13 |
| 5 | ✓ ✓ | 30.07 | 38.28 | 42.25 | 35.24 |
| 6 | ✓ ✓ | 29.83 | 38.50 | 42.51 | 35.24 |

4.3 TGP in a Zero-Shot Setting

Although TGP and Finetuning obtain significant reductions on off-target cases, they both assume some amount of direct parallel data on English-free pairs, while in reality, such direct parallel data may not exist in extreme low-resource scenario. To simulate a zero-shot setting, we build a new oracle dataset that explicitly excludes parallel data of all evaluated English-free pairs.13 In this setting, all evaluation pairs are trained in a strict zero-shot manner to test the system’s generalization ability.

Performance In Tables 5, 6, 7 row 7 with ✓ ✓, TGP in a zero-shot manner slightly lags behind TGP with full oracle data, while still gaining significant improvement compared to the baseline. On average, we observe a gain of +0.93 BLEU on En-X, +1.19 BLEU on X-En and +4.8 on English-free compared to baseline (row 7 vs. 2), and a slight

---

13We exclude the parallel data of 6 evaluation pairs from oracle data while keeping the others. More details in Section A.3.
Table 9: BLEU scores of English-free translations on OPUS-100. ✓ denotes TGP training in a zero-shot manner for all evaluated English-free pairs. As a reference, the average off-target rate reported by FastText LangID model is 4.85% on the references.

|       | TLP | TGP | De-Fr | Ru-Fr | Ni-De | Zh-Ru | Zh-Ar | Ni-Ar | BLEU | Off-Tgt Avg | Off-Tgt Avg (%) |
|-------|-----|-----|-------|-------|-------|-------|-------|-------|------|-------------|----------------|
| 1     |     |     | 18.5  | 21.5  | 21.0  | 26.7  | 21.7  | 19.7  | 13.6 | 20.2        | 14.9          | 17.8          | 16.6          | 5.7 | 18.16 | 4.2%        |
| 2     |     |     | 3.5   | 2.9   | 7.2   | 4.0   | 4.9   | 4.8   | 4.5  | 11.0        | 2.8           | 11.1          | 1.5           | 2.6 | 5.07 | 65.8%       |
| 3     |     |     | 13.1  | 14.9  | 15.9  | 17.3  | 16.0  | 14.4  | 12.8 | 14.8        | 15.2          | 12.9          | 8.5           | 3.1 | 13.24 | 3.7%        |
| 4     | ✓   | ✓   | 16.4  | 17.7  | 18.3  | 21.0  | 19.0  | 18.3  | 13.9 | 14.2        | 12.7          | 16.2          | 14.2          | 5.0 | 15.70 | 4.8%        |
| 5     | ✓   | ✓   | 4.2   | 12.2  | 12.8  | 20.9  | 10.0  | 5.9   | 11.5 | 13.5        | 12.8          | 12.6          | 11.8          | 4.5 | 11.06 | 31.1%       |
| 6     |     | ✓   | 6.6   | 14.2  | 16.7  | 21.4  | 16.2  | 8.6   | 12.9 | 14.2        | 14.7          | 12.6          | 11.8          | 4.6 | 12.88 | 16.9%       |

Table 10: Token-level off-target rates on WMT-10 quantified by whether appearing in the training set.

|       | En→X | X→En | En-Free |
|-------|------|------|---------|
| Reference | 0.10% | 0.02% | 0.73% |
| Baseline   | 0.08% | 0.02% | 0.16% |
| +TLP       | 0.08% | 0.02% | 0.09% |
| +TGP       | 0.08% | 0.02% | 0.07% |

5 Discussions on Off-Target Translations

In this section, we will discuss the off-target translation in the English-centric directions and its relationship with token-level off-targets.

5.1 Off-Targets on English-Centric Pairs

Previous literature only studies the off-target translations in the zero-shot directions. However, we show in Tables 5 and 6 that off-target translation also occurs in the English-centric directions (although to a smaller scale). Since we are using an imperfect LangID model, we quantify its error margin as the off-target rates reported on the references. We could then observe that the baseline model is producing 0.25% and 0.18% more off-target translations than the references in En-X and X-En directions respectively, which are also reduced by our proposed TLP and TGP approaches.

5.2 Token-Level Off-Targets

Given a sentence-level LangID model, we are also curious about how it represents errors at the token level. We attempt to simply quantify the token-level off-target rates by checking whether each token appears in the training data. Surprisingly, results in Table 10 show that all systems contain lower token-level off-targets than the references. We hypothesize that it is attributed to two main reasons: 1) training set also contains noisy off-target

---

14 In our case, we obtain 1k for WMT and 2k for OPUS.

15 We assume the WMT references are always in-target, since they are collected from human translators.
tokens. 2) there are domain/vocabulary mismatches between training and test set, especially for the English-free pairs.

In order to test the robustness of the FastText LangID model as well as to relate our reported sentence-level scores to the token-level, we randomly introduced off-target tokens to the references and observed the sentence-level scores. Specifically, we replaced the in-target token to a random off-target one with a probability $p$. Figure 3 shows a near exponential curve between the sentence-level scores and probability $p$. We could also observe that the sentence-level LangID model is somewhat robust to token-level off-target noises, e.g. it reports around 4% off-target rates given 20% of the tokens are replaced with off-target ones.

6 Related Work

Multilingual NMT Multilingual NMT aims to train one model to serve all language pairs (Ha et al., 2016; Firat et al., 2016; Johnson et al., 2017). Several subsequent works explored various parameter sharing strategies to mitigate the representation bottleneck (Blackwood et al., 2018; Platanios et al., 2018; Sachan and Neubig, 2018; Sen et al., 2019). Meanwhile, there are also notorious cases of off-target translation especially in English-Free pairs. Previous works either resort to back-translation techniques to generate synthetic English-Free data (Gu et al., 2019; Zhang et al., 2020), or to model-level changes to the encoder-decoder alignments (Arivazhagan et al., 2019; Liu et al., 2020). In contrast, we propose a joint representation and gradient regularization approach to reduce off-target translations and significantly improve performance across all language pairs.

Multi-Task Learning for NMT Multi-task learning (MTL) is a widely used technique to share model parameters and improve generalization (Ruder, 2017). For NMT, previous works have leveraged MTL to inject linguistic knowledge or leveraged monolingual data (Eriguchi et al., 2017; Niehues and Cho, 2017; Kiperwasser and Ballesteros, 2018; Wang et al., 2020b). Our work leverages an auxiliary TLP loss to help learn more separable model states for different target languages.

Optimization Learning Previous works have studied the optimization challenges in multi-task training (Hessel et al., 2019; Schaul et al., 2019), where Yu et al. (2020) proposed to resolve gradient conflicts between different tasks. Meanwhile for NMT, Wang et al. (2021, 2020a) proposed to mask out or assign different weights to training samples based on the gradient alignments with validation set. Yu et al. (2020) proposed to resolve the pair-wise gradient conflicts between translation directions. In contrast, we propose TGP to guide the training process by projecting training gradients according to the oracle gradients.

7 Conclusions

In this work, we aimed to reduce the off-target translations with our proposed joint representation (TLP) and gradient (TGP) regularization to guide the internal modeling of target languages. Our results showed both approaches to be highly effective at improving translation quality and to a large extent reduced the off-target occurrences. As a future direction, we will investigate off-target translations during decoding time (Yang et al., 2018, 2020).

Acknowledgements

The authors thank the anonymous reviewers for their insightful and helpful comments.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. arXiv preprint arXiv:1907.05019.
Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. ICLR.

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. arXiv preprint arXiv:1806.03280.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2019. Exploiting linguistic resources for neural machine translation using multi-task learning. arXiv preprint arXiv:1708.00993.

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

Emmanouil Antonios Plataniotis, Minminaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. arXiv preprint arXiv:1808.08493.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2020. Improving zero-shot translation by disentangling positional information. arXiv preprint arXiv:2012.15127.

Jan Niehues and Eunah Cho. 2017. Exploiting multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. arXiv preprint arXiv:1601.01073.

Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. 2019. Ray interference: a source of plateaus in deep reinforcement learning. arXiv preprint arXiv:1904.11455.

Devendra Singh Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. arXiv preprint arXiv:1809.00252.

Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. 2019. Ray interference: a source of plateaus in deep reinforcement learning. arXiv preprint arXiv:1904.11455.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Nam Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. arXiv preprint arXiv:2008.00401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.

Xinyi Wang, Ankur Bapna, Melvin Johnson, and Orhan Firat. 2021. Gradient-guided loss masking for neural machine translation. arXiv preprint arXiv:2102.13549.
Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020a. Balancing training for multilingual neural machine translation. *arXiv preprint arXiv:2004.06748*.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960.

Yiren Wang, ChengXiang Zhai, and Hany Hassan Awadalla. 2020b. Multi-task learning for multilingual neural machine translation. *arXiv preprint arXiv:2010.02523*.

Zirui Wang, Zachary C Lipton, and Yulia Tsvetkov. 2020c. On negative interference in multilingual language models. In *EMNLP*, pages 4438–4450.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020d. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. *arXiv preprint arXiv:1808.09582*.

Yilin Yang, Longyue Wang, Shuming Shi, Prasad Tadepalli, Stefan Lee, and Zhaopeng Tu. 2020. On the sub-layer functionalities of transformer decoder. *arXiv preprint arXiv:2010.02648*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. *arXiv preprint arXiv:2004.11067*.
A Appendix

A.1 WMT-10 Data
We concatenate all resources except WikiTitles from WMT2019. For Fr and Cs, we randomly sample 10M sentence pairs from the full corpus. The detailed statistics of bitext data can be found in Table 11. We randomly sample 1,000 sentence pairs from each individual validation set and concatenate them to construct a multilingual validation set.

A.2 OPUS-100 Dataset
After de-duplicating both training and dev set against the test set, the statistics of OPUS-100 dataset is shown in Table 12.

A.3 Statistics of the Oracle Data
We concatenated all available parallel dev set individually for each target language. Tables 13 and 14 illustrate the statistics of our oracle data, before and after excluding the parallel English-free data of 6 evaluation pairs.
| Code | Language | #Train | Dev  | Test  |
|------|----------|--------|------|-------|
| Fr   | French   | 10M    | Newstest13 | Newstest15 |
| Cs   | Czech    | 10M    | Newstest16 | Newstest18 |
| De   | German   | 4.6M   | Newstest16 | Newstest18 |
| Fi   | Finnish  | 4.8M   | Newstest16 | Newstest18 |
| Lv   | Latvian  | 1.4M   | Newsdev17  | Newstest17 |
| Et   | Estonian | 0.7M   | Newsdev18  | Newstest18 |
| Ro   | Romanian | 0.5M   | Newsdev16  | Newstest16 |
| Hi   | Hindi    | 0.26M  | Newsdev14  | Newstest14 |
| Tr   | Turkish  | 0.18M  | Newstest16 | Newstest18 |
| Gu   | Gujarati | 0.08M  | Newsdev19  | Newstest19 |

Table 11: Statistics of the WMT-10 dataset.

| Code | Language | Train  | Dev  | Test  |
|------|----------|--------|------|-------|
| af   | Afrikaans| 275451 | 2000 | 2000 |
| am   | Amharic  | 88979  | 2000 | 2000 |
| ar   | Arabic   | 999988 | 2000 | 2000 |
| as   | Assamese | 138277 | 2000 | 2000 |
| az   | Azerbaijani | 262060 | 2000 | 2000 |
| be   | Belarusian| 67183  | 2000 | 2000 |
| bg   | Bulgarian| 999983 | 2000 | 2000 |
| bn   | Bengali  | 999990 | 2000 | 2000 |
| br   | Breton   | 153283 | 2000 | 2000 |
| bs   | Bosnian  | 999991 | 2000 | 2000 |
| ba   | Bashkir  | 999983 | 2000 | 2000 |
| ca   | Catalan  | 999983 | 2000 | 2000 |
| cs   | Czech    | 999988 | 2000 | 2000 |
| cy   | Welsh    | 289007 | 2000 | 2000 |
| da   | Danish   | 999991 | 2000 | 2000 |
| de   | German   | 999993 | 2000 | 2000 |
| el   | Greek    | 999982 | 2000 | 2000 |
| eo   | Esperanto| 337074 | 2000 | 2000 |
| es   | Spanish  | 999996 | 2000 | 2000 |
| et   | Estonian | 999993 | 2000 | 2000 |
| eu   | Basque   | 999991 | 2000 | 2000 |
| fr   | French   | 999997 | 2000 | 2000 |
| fr   | Frisian  | 53381  | 2000 | 2000 |
| ga   | Irish    | 289339 | 2000 | 2000 |
| gd   | Gaelic   | 15689  | 1605 | 1606 |
| gl   | Galician | 515318 | 2000 | 2000 |
| gu   | Gujarati | 317723 | 2000 | 2000 |
| ha   | Hausa    | 97980  | 2000 | 2000 |
| he   | Hebrew   | 999973 | 2000 | 2000 |
| hi   | Hindi    | 534192 | 2000 | 2000 |
| hr   | Croatian | 999991 | 2000 | 2000 |
| hy   | Armenian | 289339 | 2000 | 2000 |
| id   | Indonesian| 999976 | 2000 | 2000 |
| ig   | Igbo     | 16640  | 1843 | 1843 |
| is   | Icelandic| 999980 | 2000 | 2000 |
| it   | Italian  | 999988 | 2000 | 2000 |
| ja   | Japanese | 999986 | 2000 | 2000 |
| ka   | Georgian | 377259 | 2000 | 2000 |
| kk   | Kazakh   | 79132  | 2000 | 2000 |
| km   | Central Khmer | 110924 | 2000 | 2000 |
| kn   | Kannada  | 14303  | 917  | 918  |
| ko   | Korean   | 999989 | 2000 | 2000 |
| ku   | Kurdish  | 143823 | 2000 | 2000 |
| ky   | Kyrgyz   | 25991  | 2000 | 2000 |
| li   | Limburgan| 23780  | 2000 | 2000 |
| lt   | Lithuanian| 999973 | 2000 | 2000 |

Table 12: Statistics of the OPUS-100 dataset after de-duplicating against the test set.
Table 13: Statistics of the constructed oracle data on WMT-10. Oracle° denotes after excluding the direct data of all evaluated language pairs. Language ranked by the available bilingual resources.

| Language | #Oracle | #Oracle° |
|----------|---------|----------|
| En       | 15976   | 15976    |
| Fr       | 8776    | 7180     |
| Cs       | 8776    | 7978     |
| De       | 8776    | 6383     |
| Fi       | 8776    | 8776     |
| Lv       | 8776    | 8776     |
| Et       | 8776    | 7180     |
| Ro       | 8776    | 7180     |
| Hi       | 8776    | 8776     |
| Tr       | 8776    | 7978     |
| Gu       | 8776    | 7978     |

Table 14: Statistics of the constructed oracle data on OPUS-100. Oracle° denotes after excluding the direct data of all evaluated language pairs. Language ranked alphabetically. “Others” includes all other 83 languages.

| Language | #Oracle | #Oracle° |
|----------|---------|----------|
| Ar       | 9600    | 6400     |
| De       | 9600    | 6400     |
| En       | 148427  | 148427   |
| Fr       | 9600    | 6400     |
| Gd       | 1284    | 1284     |
| Ig       | 1474    | 1474     |
| Kn       | 733     | 733      |
| Ni       | 9600    | 6400     |
| Or       | 1053    | 1053     |
| Ru       | 9600    | 6400     |
| Tk       | 1481    | 1481     |
| Zh       | 9600    | 6400     |
| Others   | 1600    | 1600     |