

Data Augmentation for Text Generation Without Any Augmented Data

Wei Bi* Huayang Li* Jiacheng Huang

Tencent AI Lab, Shenzhen, China

{victoriabi, alanili, eziohuang}@tencent.com

Abstract

Data augmentation is an effective way to improve the performance of many neural text generation models. However, current data augmentation methods need to define or choose proper data mapping functions that map the original samples into the augmented samples. In this work, we derive an objective to formulate the problem of data augmentation on text generation tasks without any use of augmented data constructed by specific mapping functions. Our proposed objective can be efficiently optimized and applied to popular loss functions on text generation tasks with a convergence rate guarantee. Experiments on five datasets of two text generation tasks show that our approach can approximate or even surpass popular data augmentation methods.

1 Introduction

End-to-end neural models are generally trained in a data-driven paradigm. Many researchers have proposed powerful network structures to fit training data well. It has also become ubiquitous to increase the training data amount to improve model performance. Data augmentation is an effective technique to create additional samples in both vision and text classification tasks (Perez and Wang, 2017; Shorten and Khoshgoftaar, 2019; Wei and Zou, 2019), which perturb samples without changing their labels. For text generation tasks, there can be more types of data perturbation to construct augmented samples, including corrupting the input text (Xie et al., 2017), the output text (Norouzi et al., 2016; Kurata et al., 2016), or both (Zhang et al., 2020). As such, classification tasks can be regarded as special cases of generation tasks in terms of incorporating data augmentation techniques, and this work mainly discusses text generation tasks.

The focus of previous work on text data augmentation has been to design proper augmentation techniques to create augmented samples. Some augmentation methods have been proposed for general text tasks. For example, different general replacement operations have been explored to edit words in a text sample, ranging from simple look-up tables (Zhang et al., 2015) to pretrained masked language models (Kobayashi, 2018; Wu et al., 2019). Sennrich et al. (2016) propose to augment text sequences by back-translation. For some generation tasks such as dialogue generation, general augmentation methods may not yield stable improvements and it requires to carefully incorporate the task property to design useful augmented samples (Zhang et al., 2020). All these methods need to explicitly construct augmented samples, and the data mapping functions from the original samples to the augmented samples are mostly defined a priori. This motivates us to raise a question, whether we can skip the step to define or choose proper augmented data mapping functions to accomplish effective data augmentation.

To answer this question, we aim to formulate the problem of data augmentation for general text generation models without any use of augmented data mapping functions. We start from a conventional data augmentation objective, which is a weighted combination of loss functions associated with the original and augmented samples. We show that the loss parts of the augmented samples can be re-parameterized by variables not dependent on the augmented data mapping functions, if a simple Euclidean loss function between the sentence representations is applied. Based on this observation, we propose to directly define a distribution on the re-parameterized variables. Then we optimize the expectation of the augmented loss parts over this distribution to approximate the original augmented loss parts computed with various augmented data

*Equal contribution.

mapping functions. We make different assumptions on the variable distributions and find that our proposed objective can be computed and optimized efficiently by simple gradient weighting. If stochastic gradient descent (SGD) is used, our objective is guaranteed with the convergence rate $O(1/\sqrt{T})$. Our objective can be coupled with popular loss functions on text generation tasks, including the word mover’s distance (Kusner et al., 2015) and the cross-entropy loss.

Our approach, which utilizes the proposed objective and optimizes it by SGD, has two advantages. First, it provides a unified formulation of various data perturbation types in general text generation models, which sheds a light on understanding the working mechanism of data augmentation. Second, the optimization of our approach is simple and efficient. Without introducing any new sample during training, we can avoid additional calculation efforts on augmented samples, often with the total size much larger than the original data size. Hence, our approach maintains high training efficiency.

Extensive experiments are conducted to validate the effectiveness of our approach. We mainly use the LSTM-based network structure (Bahdanau et al., 2015; Luong et al., 2015b) and perform experiments on two text generation tasks - neural machine translation and single-turn conversational response generation. Results on five datasets demonstrate that the proposed approach can approximate or even surpass popular data augmentation methods such as masked language model (Devlin et al., 2019) and back-translation (Sennrich et al., 2016).

2 Related Work

Data augmentation has shown promising improvements on neural models for different text generation tasks such as language modeling (Xie et al., 2017), machine translation (Sennrich et al., 2016) and dialogue generation (Niu and Bansal, 2019; Cai et al., 2020). Existing text data augmentation methods can be mainly categorized into word-level augmentation and sentence-level augmentation.

Word-level augmentation methods perturb words within the original sentence. Common operations include word insertion and deletion (Wei and Zou, 2019), synonym replacement (Zhang et al., 2015), and embedding mix-up (Guo et al., 2019). Masked language models can be used by masking some percentages of tokens at random, and predicting the masked words based on its context (Wu et al.,

2019; Cai et al., 2020).

Sentence-level data augmentation is not limited to edit only a few words in the original sentence, but to generate a complete sentence. For example, back-translation is originally proposed to translate monolingual target language data into source language to augment training pairs in machine translation (Sennrich et al., 2016). It is later extended to paraphrase sentences in any text dataset, in which two translation models are applied: one translation model from the source language to target language and another from the target to the source. GAN-based and VAE-based models have also achieved impressive results to create entire sentences to augment the training data (Hu et al., 2017; Cheng et al., 2019). For dialogue generation, retrieved sentences can be good supplement of the original corpus (Zhang et al., 2020).

Both word-level and sentence-level augmentation methods need to define their augmented data mapping functions (i.e. operations to edit words or models to generate sentences) apriori. Some works train policies to sample a set of word-level operations (Niu and Bansal, 2019), but the operation candidates are still pre-defined. A few works learn to construct augmented samples and optimize the network jointly (Hu et al., 2019; Cai et al., 2020). Different from previous work, our goal is not to propose or learn novel augmented data mapping functions. Instead, we investigate whether the effectiveness of data augmentation can be achieved while we do not bother to use any specific augmented data mapping function.

Besides data augmentation, data weighting is another useful way to improve model learning. It assigns a weight to each sample to adapt its importance during training. The sample weights are often carefully defined (Freund and Schapire, 1997; Bengio et al., 2009) or learnt by another network (Jiang et al., 2018; Shu et al., 2019). Data augmentation is often combined with data weighting together to weight the original and augmented samples.

3 Background

We are given original samples $\mathcal{D} = \{(x, y)\}$ with x, y both as text sequences. Without loss of generality, a deep generation model is to learn a mapping function $f_{x,y}$ by a deep neural network that outputs y given x . As mentioned in the introduction, text generation tasks mainly have three types of augmented data:

- one (or several) perturbed input text \hat{x} by one (or several) augmented data mapping function $\phi_{\hat{x}}$;
- one (or several) perturbed output text \hat{y} by one (or several) augmented data mapping functions $\phi_{\hat{y}}$;
- one (or several) perturbed paired text (\hat{x}, \hat{y}) by corresponding augmented data mapping functions.

Proper augmented data mapping functions are often supposed to generate perturbed sequences or sequence pairs that are close to the original one. They are assumed to be given apriori in optimizing the generation model for now.

Let $\ell(f_{x,y}(\mathbf{x}), \mathbf{y})$ denote the loss function to be minimized for each sample. We first use augmented data in the input domain as an example to present the problem formulation and introduce our approach, then later discuss other types of augmented data. Data augmentation methods generally apply an augmented loss per sample with its augmented samples:

$$\ell_{aug} = \ell(f_{x,y}(\mathbf{x}), \mathbf{y}) + \sum_{\hat{x}: \phi_{\hat{x}} \in \mathcal{F}} w_{\hat{x}} \ell(f_{x,y}(\hat{x}), \mathbf{y}) \quad (1)$$

where $w_{\hat{x}}$ is the importance weight associated with each augmented sample, $\phi_{\hat{x}}$ is the augmented data mapping function that constructs \hat{x} , and \mathcal{F} is the function space containing all feasible augmented data mapping functions.

4 Our Approach

In this section, we aim to formulate the problem of data augmentation for general text generation models without any use of augmented data mapping functions. We introduce our approach by assuming that the loss function ℓ is the most simple Euclidean distance, i.e.

$$\ell(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2 \quad (2)$$

where \mathbf{u} and \mathbf{v} are the sentence representations of two sentences, i.e. the target sequence and the predicted sequence. Other conventional loss functions in text generation will be discussed in Section 5.

We first rewrite each loss part of an augmented data point in (1) from a polar coordinate system in Sec 4.1. In this way, we can regard the total augmented loss part with multiple augmented data mapping functions as sampling different points in the polar coordinate system. This inspires us that we can skip to define any augmented data mapping function, but only design a joint distribution of the

perturbation radius and perturbation angle in the polar coordinate system. In Sec 4.2, we show two probability distribution substantiations, and find that our approach can be optimized efficiently by simply re-weighting the gradients. In Sec 4.3, we discuss the extension of our approach for other augmented data mapping function types.

4.1 Proposed Objective

By treating $f_{x,y}(\mathbf{x})$, $f_{x,y}(\hat{x})$ and \mathbf{y} as three vertices in the Euclidean space, we can form a triangle (illustrated in Fig. 1a) with the three vertices and the loss between them as edges. For a given augmented data mapping function $\phi_{\hat{x}}$ and a sample (\mathbf{x}, \mathbf{y}) , we can rewrite $\ell(f_{x,y}(\hat{x}), \mathbf{y})$ using the polar coordinate system with $f_{x,y}(\mathbf{x})$ as the pole and $(f_{x,y}(\mathbf{x}), \mathbf{y})$ as the polar axis:

$$\begin{aligned} \ell^2(f_{x,y}(\hat{x}), \mathbf{y}) = & \\ & \ell^2(f_{x,y}(\mathbf{x}), \mathbf{y}) + \ell^2(f_{x,y}(\mathbf{x}), f_{x,y}(\hat{x})) \\ & - 2\ell(f_{x,y}(\mathbf{x}), f_{x,y}(\hat{x}))\ell(f_{x,y}(\mathbf{x}), \mathbf{y}) \cos \theta \end{aligned} \quad (3)$$

where θ is the radian of $f_{x,y}(\hat{x})$. We can observe that, the rewritten augmented sample loss part depends on the original sample loss $\ell(f_{x,y}(\mathbf{x}), \mathbf{y})$ as well as the radius r and radian θ of $f_{x,y}(\hat{x})$. Here r is the data perturbation distance $\ell(f_{x,y}(\mathbf{x}), f_{x,y}(\hat{x}))$. Therefore, we can map each augmented data mapping function $\phi_{\hat{x}} \in \mathcal{F}$ into $(r, \theta) \in P$, where P is a joint distribution of (r, θ) ¹. A weighted summation of the augmented loss parts from different augmented data mapping functions can be seen as an empirical estimation of the expectation of the rewritten loss by sampling different (r, θ) 's from their joint distribution P , though the corresponding ground truth P is not observed.

This inspires us how to avoid to specifically design or choose several augmented data mapping functions and their weights used in (1). We can directly design the distribution P of (r, θ) and optimize the expectation of the rewritten loss (i.e. the right hand side in (3)) under this distribution. Hence, we propose to optimize the following objective to mimic the effect of data augmentation:

¹It is worth pointing out that even if the three vertices (i.e., $f_{x,y}(\hat{x})$, \mathbf{y} , and $f_{x,y}(\mathbf{x})$) lie in high dimensional spaces, we can always use the distribution of (r, θ) cover all possible triangles formed by them. And our derivation will not lose its generalization in high dimensional spaces, since we does not make use of the vertices but only edges of the triangles.

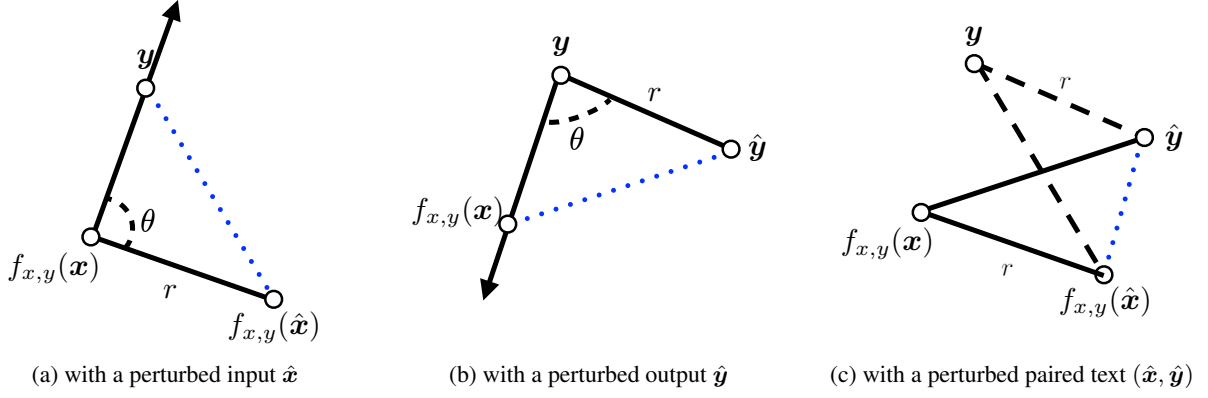


Figure 1: Illustration of the polar coordinate systems for three kinds of data perturbation. Rays in the figures are the polar axes. Our approach expresses edges in dots by their corresponding polar coordinates.

$$\ell_{our} = \ell(f_{x,y}(\mathbf{x}), \mathbf{y}) + \mathbb{E}_{(r,\theta) \in P} [\Phi(\ell(f_{x,y}(\mathbf{x}), \mathbf{y}))] \quad (4)$$

where $\Phi(e; r, \theta)$ is a function of an edge e in the loss function space given (r, θ) :

$$\Phi(e; r, \theta) = \sqrt{e^2 + r^2 - 2er \cos \theta}. \quad (5)$$

4.2 Optimization

We design specific distributions of (r, θ) used in the proposed objective (4) and their optimization. We assume the two variables are independent:

$$p(r, \theta) = p(r)p(\theta). \quad (6)$$

In the following corollary, we first show the result by assuming that both r and θ follow uniform distributions. Recall that proper data mapping functions augment samples close to the original one. An ideal case is thus to perturb samples with their output representations uniformly surrounding that of the original sample. The uniform distribution with a small perturbation radius upper bound R can simulate this ideal case.

Corollary 1. *We are given the perturbation distance upper bound R and assume that*

$$r \sim \mathcal{U}(0, R), \theta \sim \mathcal{U}(0, \pi). \quad (7)$$

$\mathbb{E}_{(r,\theta) \in P} [\Phi(\ell(f_{x,y}(\mathbf{x}), \mathbf{y}))]$ is upper bounded by $\frac{1}{2}\ell(f_{x,y}(\mathbf{x}), \mathbf{y}) + C_1 \cdot \ell^2(f_{x,y}(\mathbf{x}), \mathbf{y}) + C_2(R)$, where C_1 is a constant and $C_2(R)$ is another constant dependent on R .

Proof is in the Appendix. With the above result, we can optimize the objective in (4) by minimizing the derived upper bound. We calculate its gradient:

$$\frac{\partial \ell_{our}}{\partial \Theta} = \frac{3}{2} \cdot \frac{\partial \ell(\Theta)}{\partial \Theta} + 2C_1 \cdot \ell(\Theta) \frac{\partial \ell(\Theta)}{\partial \Theta} \quad (8)$$

where Θ contains all neural model parameters. It can be observed that the major difference of the above gradient compared with the original one of the objective in (1) lies in the second part of (8), which weights the original gradient by the loss value. This means that the performance improvement brought by data augmentation under our formulation can be equivalently accomplished by specialized data weighting. Indeed, many data weighting methods (Lin et al., 2017) favors hard examples by reducing the gradient contribution from easy examples and increasing the importance of hard examples (example with large loss value in our approach), which significantly boost the performance. This in turn shows that simple uniform distributions assumed here should be reasonable and effective.

Instead of uniform distribution, we can assume a uniform distribution on θ but an exponential distribution on r such that a small perturbation distance is preferred with a higher probability.

Corollary 2. *We are given the expected value of the perturbation distance as R and assume that*

$$r \sim \text{Exp}\left(\frac{1}{R}\right), \theta \sim \mathcal{U}(0, \pi). \quad (9)$$

$\mathbb{E}_{(r,\theta) \in P} [\Phi(\ell(f_{x,y}(\mathbf{x}), \mathbf{y}))]$ is upper bounded by $C_1(R) \cdot \ell(f_{x,y}(\mathbf{x}), \mathbf{y}) + \frac{C_1(R)}{2} \cdot \ell^2(f_{x,y}(\mathbf{x}), \mathbf{y}) + C_2(R)$, where $C_1(R)$ and $C_2(R)$ are constants dependent on R .

Proof is in the Appendix. The above corollary shows that even if different distributions are assumed, we can still use gradient weighting to optimize the proposed objective, where $C_1(R)$ can be set as a hyper-parameter.

If the loss is Lipschitz smooth, of which Euclidean distance is the case, we can prove the convergence of our approach with the convergence rate

$O(1/\sqrt{T})$, if SGD is used. The proof is provided in the Appendix, which is extended from results in Reddi et al. (2016).

Theorem 1. *Suppose ℓ_{our} is in the class of finite-sum Lipschitz smooth functions, has δ -bounded gradients, and the weight of the loss gradient is clipped to be bounded by $[w_1, w_2]$. Let the learning rate of SGD $\alpha_t = c/\sqrt{T}$ where $c = \sqrt{\frac{2(\ell_{our}(\Theta^0) - \ell_{our}(\Theta^*))}{L\sigma^2 w_1 w_2}}$ where L is the Lipschitz constant and Θ^* is an optimal solution. Then the iterates of SGD of our approach with ℓ_{our} satisfy:*

$$\min_{0 \leq t \leq T-1} \mathbb{E}[\|\nabla \ell_{our}(\Theta^t)\|^2] \leq \sqrt{\frac{2(\ell_{our}(\Theta^0) - \ell_{our}(\Theta^*))Lw_1}{Tw_2}} \sigma. \quad (10)$$

4.3 Other Types of Augmented Data

We now discuss how our approach can be applied to other types of augmented data. For augmented data on the output domain, the objective in (1) becomes:

$$\ell_{aug} = \ell(f_{x,y}(\mathbf{x}), \mathbf{y}) + \sum_{\phi_{\hat{\mathbf{y}}} \in \mathcal{F}} w_{\hat{\mathbf{y}}} \ell(f_{x,y}(\mathbf{x}), \hat{\mathbf{y}}). \quad (11)$$

The augmented loss part can be rewritten using the polar coordinate system with \mathbf{y} as the pole and $(\mathbf{y}, f_{x,y}(\mathbf{x}))$ as the polar axis, illustrated in Fig. 1b:

$$\begin{aligned} \ell^2(f_{x,y}(\mathbf{x}), \hat{\mathbf{y}}) &= \ell^2(\mathbf{y}, f_{x,y}(\mathbf{x})) + \ell^2(\mathbf{y}, \hat{\mathbf{y}}) \\ &\quad - 2\ell(\mathbf{y}, f_{x,y}(\mathbf{x}))\ell(\mathbf{y}, \hat{\mathbf{y}}) \cos \theta. \end{aligned} \quad (12)$$

Similarly, the augmented data mapping function $\phi_{\hat{\mathbf{y}}}$ can be re-parameterized into a function of the radius $r = \ell(\mathbf{y}, \hat{\mathbf{y}})$ (still the perturbation distance) and the radian of $\hat{\mathbf{y}}$. The objective turns out to be the same as (4).

For data perturbation on both the input and output space, we have:

$$\ell_{aug} = \ell(f_{x,y}(\mathbf{x}), \mathbf{y}) + \sum_{\phi_{\hat{\mathbf{x}}, \hat{\mathbf{y}}} \in \mathcal{F}} w_{\hat{\mathbf{x}}, \hat{\mathbf{y}}} \ell(f_{x,y}(\hat{\mathbf{x}}), \hat{\mathbf{y}}). \quad (13)$$

Illustrated in Fig. 1c, we first make use of the triangle inequality that:

$$\begin{aligned} \ell(f_{x,y}(\hat{\mathbf{x}}), \hat{\mathbf{y}}) &\leq \frac{1}{2}(\ell(f_{x,y}(\hat{\mathbf{x}}), \mathbf{y}) + \ell(\mathbf{y}, \hat{\mathbf{y}})) \\ &\quad + \frac{1}{2}(\ell(f_{x,y}(\hat{\mathbf{x}}), f_{x,y}(\mathbf{x})) + \ell(f_{x,y}(\mathbf{x}), \hat{\mathbf{y}})). \end{aligned} \quad (14)$$

Using (3) and (12), the objective is rewritten as:

$$\begin{aligned} \ell_{our} &= \ell(f_{x,y}(\mathbf{x}), \mathbf{y}) \\ &\quad + \mathbb{E}_{(r,\theta) \in P} [r + \Phi(\ell(f_{x,y}(\mathbf{x}), \mathbf{y}))]. \end{aligned} \quad (15)$$

Note that $\mathbb{E}_{(r,\theta) \in P} [r]$ is a scalar which is not dependent on any learning parameter. Thus optimizing the above objective is equivalent to optimizing (4).

From the above analysis, we can see that our proposed objective in (4) can be applied to handle all three kinds of augmented data mapping functions in text generation models.

5 Loss Function

In theory, our approach can be applied to any Lipschitz smooth loss function that holds the equation (3). In this section, we show another valid loss function in our approach – the word mover’s distance (WMD) (Kusner et al., 2015; Zhao et al., 2019), which is previously used in various text generation tasks. Next, we discuss the cross entropy loss, in which the proposed objective is not an upper-bound of the data augmentation objective. However, our approach can still converge with the same convergence rate and experimental results in the next section validate the effectiveness of our approach with the cross-entropy loss.

5.1 Word Mover’s Distance

WMD, also named the optimal transport distance (Chen et al., 2018a), leverages optimal transport to find an optimal matching of similar words between two sequences, providing a way to measure their semantic similarity:

$$\begin{aligned} \ell_{WMD}(\mathbf{u}, \mathbf{v}) &= \min_{T_{i,j}} \sum_{i,j} T_{i,j} d_{i,j} \quad (16) \\ \text{s.t.} \quad &\sum_{j=1}^M T_{i,j} = p_{u,i} \quad \forall i \\ &\sum_{i=1}^N T_{i,j} = p_{v,j} \quad \forall j \end{aligned}$$

where $p_{u,i}/p_{v,j}$ is the probability distribution of the sentence, i.e. $\sum_i p_{u,i} = 1$ and $\sum_j p_{v,j} = 1$. $d_{i,j}$ is the cost for mis-predicting u_i to v_j , where the squared Euclidean distance $d_{i,j} = \|u_i - v_j\|^2$ is used and u_i/v_j is the word embedding vector. Note that the Euclidean distance in (2) is a special case of WMD by replacing the 1-gram used in WMD

to n -gram with n larger than the sentence’s length. WMD is the squared L^2 Wasserstein distance. We take its squared root, i.e. $\ell_{WD} = \sqrt{\ell_{WMD}}$, which holds an upper bound as the right hand side in (3). Also, ℓ_{WD} is Lipschitz smooth.

Theorem 2. *For the L^2 Wasserstein distance $W_2(\cdot, \cdot)$ on the Wasserstein space $W^2(\mathbb{R}^n)$ and any $x, y, z \in W^2(\mathbb{R}^n)$, we have*

$$W_2(y, z)^2 \leq W_2(x, y)^2 + W_2(z, x)^2 - 2 \cdot W_2(x, y) \cdot W_2(z, x) \cdot \cos \theta. \quad (17)$$

Here θ is the angel between the γ_{xy} and γ_{zx} , γ_{xy} is the geodesic (shortest path) connecting x, y in $W^2(\mathbb{R}^n)$, and γ_{zx} is the geodesic connecting z, x in $W^2(\mathbb{R}^n)$.

Theorem 3. *u and v are given as fixed. Assuming that \mathbf{u}_Θ is Lipschitz continuous with respect to the parameters Θ . Then $\ell_{WD}(\mathbf{u}_\Theta, \mathbf{v})$ is Lipschitz continuous with respect to the parameters Θ .*

Roughly speaking, according to Sturm et al. (2006)[Proposition 2.10], the sectional curvature of Wasserstein space $W^2(\mathbb{R}^n)$ is non-negative. Hence, every geodesic triangle in $W^2(\mathbb{R}^n)$ is fatter than the one with same sides length in \mathbb{R}^2 . As a consequence, an inequality like cosine law is satisfied on $W^2(\mathbb{R}^n)$, i.e., Theorem 2 holds. A formal proof of the above two theorems is provided in the Appendix. Thus, all our derivations in Section. 4 hold.

The exact computation of ℓ_{WD} is expensive during training. In our experiments, we resort to the inexact proximal point method for optimal transport algorithm to compute it (Chen et al., 2018a).

5.2 Cross-entropy Loss

Although WMD is effective for various sequence generation tasks, the most conventional loss function adopted in existing generation models is the cross-entropy loss. It measures the word difference at each word y_i of the output sequence \mathbf{y} :

$$\ell_{CE}(\mathbf{y}_i, \mathbf{p}_i) = \mathbf{y}_i^T \log(\mathbf{p}_i) \quad (18)$$

$$\ell_{CE}(\mathbf{y}, \mathbf{p}) = \sum_{i=1}^{|\mathbf{y}|} \ell_{CE}(\mathbf{y}_i, \mathbf{p}_i) \quad (19)$$

where \mathbf{y}_i is the target one-hot vector with the correct dimension as 1 and 0 elsewhere, and \mathbf{p}_i is the predicted probability output by a softmax layer. We adopt the maximum likelihood estimation as the training paradigm by assuming truth for preceding words in predicting \mathbf{p}_i .

The cross-entropy loss is also Lipschitz smooth, and thus we can guarantee its convergence from Theorem 1. Unfortunately, it does not satisfy the equation in (3), and thus minimizing our objective in (4) does not necessarily approximate the data augmentation objective in (1). In our experiments, we also try the cross-entropy loss, and results show that our objective is effective to improve the model performance compared with the base model. This is not surprising since our approach is optimized by gradient weighting and thus at least it is a useful data weighting method.

6 Experiments

The proposed approach provides a new paradigm and understanding of data augmentation for text generation. To evaluate that our approach can mimic the effect of data augmentation, we conduct experiments on two text generation tasks – neural machine translation and conversational response generation. We compare our approach with two most popular data augmentation methods (one token-level and one sentence-level augmentation method) that can be applied on various text generation tasks:

- Masked Language model (MLM): We use a pre-trained BERT (Devlin et al., 2019; Wolf et al., 2020) and randomly choose 15% of the words for each sentence. BERT takes in these masked words to predict these masked positions with new words. We augment one sample from each original training sample. Thus the data size increases to twice of the original one. Note that we only augment the English side of translation datasets.
- Back-translation (BT): For neural machine translation, we employ a fixed target-to-source translation model trained on the original dataset. For conversational response generation, we perturb both the input and output text of the original sample pair using two pretrained translation model: an English-to-German model and its backward counterpart, which are obtained using the WMT14 corpus with 4.5M sentence pairs². We again augment one sample from each original training sample.

We set the same weight w of all augmented loss parts used in ℓ_{aug} as a hyper-parameter, and tune it on the development set of each dataset. Since Euclidean distance is a special case of WMD as dis-

²Datasets used in this work can be found at <https://nlp.stanford.edu/projects/nmt/>, <http://coai.cs.tsinghua.edu.cn/hml/dataset/#commonsense>

Model	De⇒En	En⇒De	Vi⇒En	En⇒Vi	Fr⇒En	En⇒Fr	It⇒En	En⇒It
CE	27.98	22.85	24.22	27.09	<u>40.49</u>	40.86	29.70	26.85
CE+MLM	28.70	23.23	24.40	26.20	40.03	40.79	29.35	26.90
CE+BT	29.35	24.09	25.00	27.41	40.87	42.64	30.44	27.94
CE+OURS	<u>29.16</u>	<u>23.26</u>	<u>24.74</u>	<u>27.12</u>	40.46	<u>40.94</u>	<u>29.79</u>	<u>27.11</u>
WD	28.53	22.95	24.03	26.69	39.71	40.48	29.74	27.08
WD+MLM	<u>28.80</u>	22.98	<u>24.33</u>	26.88	39.57	<u>40.61</u>	29.98	26.59
WD+BT	28.56	<u>23.10</u>	24.51	<u>26.74</u>	<u>39.77</u>	40.60	29.56	27.33
WD+OURS	28.91	23.42	24.26	26.73	40.46	41.07	<u>29.86</u>	<u>27.15</u>

Table 1: BLEU scores on various translation datasets. CE: Cross-Entropy loss; WD: L^2 Wasserstein distance. The best results are in **bold**, and the second-best results are in underline.

cussed in Sec 5.1, we show results of all methods with the use of the cross-entropy loss and WD. We mainly use the Fairseq (Ott et al., 2019) Seq2seq implementation as our model. Both encoder and decoder are one-layer LSTM. The word embedding dimension is 256. Attention (Luong et al., 2015b) is used with a dropout rate of 0.1. All parameters are randomly initialized based on the uniform distribution $[-0.1, +0.1]$. We use SGD to optimize our models, and the learning rate is started with 1.0. After 8 epochs, we start to halve the learning rate after each epoch. All experiments are run on a single NVIDIA V100 GPU. Code for our experiments are available once our work is accepted.

6.1 Neural Machine Translation

We use translation benchmarks IWSLT14 En-De, En-Fr, En-It, and IWSLT15 En-Vi in our experiments. The datasets of IWSLT14 are pre-processed with the script in Fairseq³. For IWSLT14 datasets, we use tst2011 as validation set and tst2012 as test set. The IWSLT15 dataset is the same as that used in Luong et al. (2015a), and the validation and test sets are tst2012 and tst2013, respectively.

Table 1 shows the BLEU scores on their test sets. For both cross-entropy loss and L^2 Wasserstein distance, all data augmentation methods (MLM, BT and OURS) perform better than the corresponding base models in most cases. The improvement margins are different across the various datasets. The reason may be that the datasets are in different scales and the alignment difficulty between different languages can also vary. The performance of MLM is not stable from our results, which is largely due to that masked tokens are possible to

³<https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-iwslt14.sh>

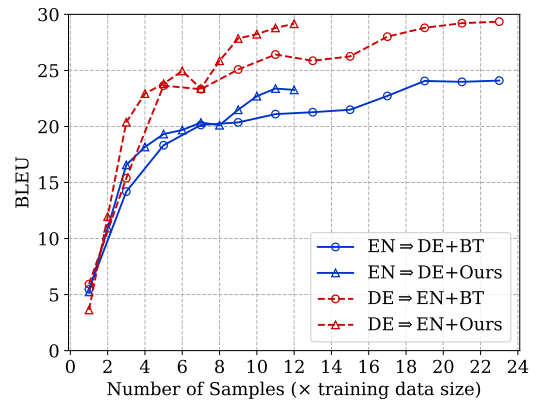


Figure 2: BLEU scores by models updated with the same number of samples.

be filled in with different semantic ones and thus the semantics of the sentence changes. Therefore, the augmented data are not aligned indeed, and the translation model learning can be distracted. Note that we also evaluate our method using the Transformer model and get some similar findings. Experimental results of the Transformer model are presented in the appendix.

Compared to BT and MLM, our approach that mimics the effect of data augmentation without actually constructing augmented samples, shows encouraging results. Note that our proposed objective may not have a theoretical guarantee on the cross-entropy loss. Yet, it still manages to improve the base model except for Fr⇒En, and surpasses MLM on all datasets. With the use of L^2 Wasserstein distance, our approach even outperforms BT and achieves the best performance on half test sets. This validates the benefits of not using any specific data augmentation mapping function in data augmentation as in our proposed objective.

Model	PPL	BLEU	BLEU-1	BLEU-2	Dist1	Dist2	Flu	Rel
CE	7.22	0.75	16.35	<u>1.38</u>	0.889	0.855	<u>3.571</u>	<u>3.314</u>
CE+MLM	6.82	<u>0.76</u>	<u>16.65</u>	1.31	0.917	0.868	3.552	3.184
CE+BT	7.38	0.68	17.04	1.33	0.892	0.851	3.557	3.249
CE+OURS	<u>7.10</u>	0.85	16.41	1.44	0.894	0.864	3.632	3.370
WD	7.10	0.87	15.09	<u>1.33</u>	0.872	<u>0.863</u>	3.644	<u>3.354</u>
WD+MLM	7.09	0.57	15.75	1.25	0.913	0.881	3.575	3.188
WD+BT	6.92	0.81	<u>15.97</u>	1.29	0.881	0.853	3.579	3.279
WD+OURS	<u>7.01</u>	<u>0.84</u>	16.56	1.39	<u>0.893</u>	0.855	<u>3.629</u>	3.447
HUMAN	-	-	-	-	0.947	0.897	4.235	4.086

Table 2: Automatic and human evaluation results on Reddit. Human: the gold reference of the query. The best results are in **bold**, and the second-best results are in underline.

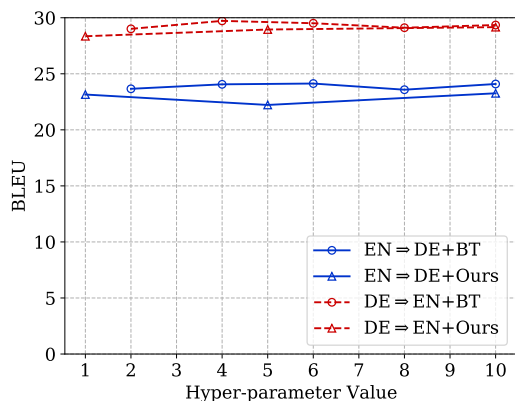


Figure 3: BLEU scores by models trained with different hyper-parameters. Values in the x-axis are re-scaled in order to visualize them in the same range.

We provide further analysis on the performance of our approach versus BT. In Fig. 2, we compare testing BLEU scores obtained by models updated with the same number of samples. Since we construct one augmented sample from each original training sample, the total number of samples used in BT is twice as much as that of our approach. We can see that our approach achieves compatible performance with BT, while only requires half of the training data. This shows that our approach, without involving additional calculations on extra samples, can effectively save the computational expense. Fig. 3 shows the sensitivity of performance under different hyper-parameters. For our approach, we vary across different $C_1(R)$'s; for BT, we vary the sample weight w of the augmented samples. We re-scale $C_1(R)$ by 10^{-4} and w by 10^{-1} , in order to visualize them within the same

range of x-axis. Both BT and our approach demonstrate their robustness under different settings of their hyper-parameters.

6.2 Conversational Response Generation

We use the English single-round Reddit conversation dataset (Zhou et al., 2018). Following previous work on data augmentation for dialogue system (Cai et al., 2020; Zhang et al., 2020), we simulate a low data regime so that data augmentation is expected to be more effective. Thus, we select data pairs with the length of both the query and response less than 20, and randomly split them into 200K for training, 2K for validation and 5K for testing. Automatic evaluation for each method is performed on all test data. We report Perplexity, BLEU and BLEU-k ($k=1,2$) to measure the response coherence; Distinct-k ($k=1,2$) (Li et al., 2016) to measure the response diversity. We also hire five annotators from a commercial annotation company for manual evaluation on 200 pairs randomly sampled from the test set. Results of all methods are shuffled for annotation fairness. Each annotator rates each response on a 5-point scale (1: not acceptable; 3: acceptable; 5: excellent; 2 and 4: used in unsure case) from two perspectives: Fluency and Relevance.

Results are summarized in Table 2. On automatic metrics, BT only shows marginal improvements on a few metrics, which can not exhibit its strength as in translation tasks. MLM effectively increases the response diversity (Dist1&2). This is due to nature of the conversation data that conversation pair often remains coherent even if the semantics of the query or response has been slightly

changed. Thus, MLM can increase data diversity, which is appreciated in training response generation models. In terms of human evaluation, BT and MLM can barely improve the base model. As for our approach, it achieves the best or second best results on most metrics for both loss functions, demonstrating more robust performance than BT and MLM. This is consistent with our statement in the introduction that we often need to design proper augmented data mapping functions carefully for a target generation task, which requires non-trivial work. As such, it is meaningful to avoid the use of specific data augmentation techniques and find a unified formulation of data augmentation for general generation tasks. From our results, the proposed objective demonstrates its power to achieve the effect of data augmentation across different generation tasks.

7 Conclusions and Future Work

We have proposed an objective of formulating data augmentation without any use of any augmented data mapping function. We show its optimization and provide the corresponding convergence rate. Both the L^2 Wasserstein distance and the cross-entropy loss are discussed with their use in our objective and their corresponding theoretical guarantees. Different from previous data augmentation works that need to add manipulated data into the training process, our gradient based approach provides a potential way to obtain performance improvements, which may come from augmented data, without incurring the computational expense. Experiments on both neural machine translation and conversational response generation validate the effectiveness of our objective compared to existing popular data augmentation methods: masked language models and back-translation.

We believe this work provides a new understanding of data augmentation. Our approach can also be useful to a wide range of tasks including text classification tasks, which can be seen as special cases of text generation tasks, and cross-modality generation tasks such as image captioning, in which we can skip the step to use various image augmentation techniques.

We would like to point out that some parts of our approach can be improved in the future, which may lead to a better performance and generalization. Firstly, current distributions we choose in the re-parameterized loss are relatively simple. Some

points under current continuous distributions may not correspond to valid text sequences in the original text space, due to the discreteness of natural languages. A possible way is that we change to leverage more informative distributions, such as including prior distributions computed from several augmented samples. Secondly, our method is derived under the framework of SGD and it is possible to extend it to the Adam framework (Kingma and Ba, 2014; Chen et al., 2018b; Reddi et al., 2019). We also leave the more general version of our work in the future.

References

- Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 41–48.
- Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6334–6343.
- Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2018a. Improving sequence-to-sequence learning via optimal transport. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. 2018b. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*.
- Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4324–4333.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.
- Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an

- application to boosting. *Journal of computer and system sciences*, 55(1):119–139.
- Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.
- Zhiting Hu, Bowen Tan, Russ R Salakhutdinov, Tom M Mitchell, and Eric P Xing. 2019. Learning data manipulation for augmentation and weighting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15764–15775.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1587–1596.
- Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2304–2313.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 452–457.
- Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Labeled data generation with encoder-decoder lstm for semantic slot filling. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, pages 725–729.
- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 957–966.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 110–119.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.
- Minh-Thang Luong, Christopher D Manning, et al. 2015a. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 76–79.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421.
- Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1317–1323.
- Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems (NeurIPS)*, pages 1723–1731.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.
- Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. 2016. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 314–323.
- Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. 2019. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 86–96.
- Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.
- Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1919–1930.

- Karl-Theodor Sturm et al. 2006. On the geometry of metric measure spaces. *Acta mathematica*, 196(1):65–131.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in the Neural Information Processing Systems (NeurIPS)*, pages 6000–6010.
- Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 38–45.
- Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Proceedings of the International Conference on Computational Science (ICCS)*, pages 84–95.
- Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Rongsheng Zhang, Yinhe Zheng, Jianzhi Shao, Xiaoxi Mao, Yadong Xi, and Minlie Huang. 2020. Dialogue distillation: Open-domain dialogue augmentation using unpaired data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3449–3460.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 28:649–657.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 563–578.
- Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 4623–4629.

A Proof of Corollary 1

$$\begin{aligned}
& E_{(r,\theta) \in P}[\sqrt{L^2 + r^2 - 2Lr \cos \theta}] \\
&= \int_{r=0}^R \int_{\theta=0}^{\pi} \frac{1}{R} \cdot \frac{1}{\pi} \cdot \sqrt{L^2 + r^2 - 2Lr \cos \theta} dr d\theta, \\
&= \int_{r=0}^R \frac{1}{R} \cdot \frac{1}{\pi} \left(\int_{\theta=0}^{\pi/2} \sqrt{L^2 + r^2 - 2Lr \cos \theta} d\theta + \int_{\theta=\pi/2}^{\pi} \sqrt{L^2 + r^2 - 2Lr \cos \theta} d\theta \right) dr \\
&\leq \int_{r=0}^R \frac{1}{R} \frac{1}{2} (\sqrt{L^2 + r^2} + L + r) dr \\
&= \frac{1}{2}L + \frac{R}{4} + \frac{1}{2R} \int_{r=0}^R \sqrt{L^2 + r^2} dr \\
&\leq \frac{1}{2}L + \frac{R}{4} + \frac{1}{2R} \int_{r=0}^R \frac{1 + L^2 + r^2}{2} dr \\
&= \frac{1}{2}L + L^2 C_1 + C_2(R). \tag{20}
\end{aligned}$$

where $L = \ell(f_{x,y}(x), y)$, $C_1 = \frac{1}{4}$, $C_2(R) = \frac{R^2}{12} + \frac{R}{4} + \frac{1}{4}$.

B Proof of Corollary 2

$$\begin{aligned}
& \int_{r=0}^{\infty} \int_{\theta=0}^{\pi} \frac{1}{R} \exp\left(-\frac{r}{R}\right) \frac{1}{\pi} (\sqrt{L^2 + r^2 - 2Lr \cos \theta}) dr d\theta \\
&\leq \int_{r=0}^R R \exp\left(-\frac{r}{R}\right) \frac{1}{2} (\sqrt{L^2 + r^2} + L + r) dr \\
&= \int_{r=0}^R R \exp\left(-\frac{r}{R}\right) \frac{1}{2} (L + r) dr + \int_{r=0}^R R \exp\left(-\frac{r}{R}\right) \frac{1}{2} (\sqrt{L^2 + r^2}) dr \\
&= \frac{R^2}{2} (1 - e^{-1})L + \frac{R^3}{2} (1 - 2e^{-1}) + \int_{r=0}^R R \exp\left(-\frac{r}{R}\right) \frac{1}{2} (\sqrt{L^2 + r^2}) dr \tag{21}
\end{aligned}$$

$$\begin{aligned}
&\leq \frac{R^2}{2} (1 - e^{-1})L + \frac{R^3}{2} (1 - 2e^{-1}) + \int_{r=0}^R R \exp\left(-\frac{r}{R}\right) \frac{1 + L^2 + r^2}{4} dr \tag{22} \\
&= LC_1(R) + L^2 \frac{C_1(R)}{2} + C_2(R)
\end{aligned}$$

where $C_1(R) = (1 - e^{-1})\frac{R^2}{2}$, and $C_2(R) = \frac{R^3}{2} + \frac{3R^2}{4} - (\frac{R^3}{2} + \frac{R^4}{2})e^{-1}$.

C Proof of Theorem 1

We study the nonconvex *finite-sum* problems of the form

$$\min_{\Theta} \mathcal{L}(\Theta) := \frac{1}{n} \sum_{i=1}^n \ell_{our}(\Theta, x_i, y_i), \tag{23}$$

where both \mathcal{L} and ℓ_{our} may be nonconvex. For ease of notation, we use ℓ to denote ℓ_{our} in the following of the proof. We denote the class of such finite-sum Lipschitz smooth functions by \mathcal{F}_n . We optimize functions in \mathcal{F}_n with the gradient in Eq. 8 by SGD. For $\mathcal{L} \in \mathcal{F}_n$, SGD takes an index $i \in [n]$ and a sample in the training set, and returns the pair $(\ell_i(\Theta), \nabla \ell_i(\Theta))$.

Definition 1. We say $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}$ is L -smooth if there is a constant L such that

$$\|\nabla \ell(\Theta') - \nabla \ell(\Theta)\| \leq L \|\Theta' - \Theta\|, \forall \Theta', \Theta \in \mathbb{R}^d. \tag{24}$$

Definition 2. A point Θ is called ϵ -accurate if $\|\nabla\ell(\Theta)\|^2 \leq \epsilon$. A stochastic iterative algorithm is said to achieve ϵ -accuracy in t iterations if $\mathbb{E}[\|\nabla\ell(\Theta^t)\|^2] \leq \epsilon$, where the expectation is over the stochasticity of the algorithm.

Definition 3. We say $\ell \in \mathcal{F}_n$ has σ -bounded gradients if $\|\nabla\ell_i(\boldsymbol{\theta})\| \leq \sigma$ for all $i \in [n]$ and $\Theta \in \mathbb{R}^d$.

Let α_t denote the learning rate at iteration t , and w_{i_t} be the gradient weight assigned to sample i by our approach. By SGD, we have

$$\Theta^{t+1} = \Theta^t - \alpha_t w_{i_t} \nabla\ell_{i_t}(\Theta^t), i \in [n]. \quad (25)$$

Definition 4. We say the positive gradient weight w in our approach is bounded if there exist constants w_1 and w_2 such that $w_1 \leq w_i \leq w_2$ for all $i \in [n]$.

Proof of Theorem 1. According to the Lipschitz continuity of $\nabla\ell$, the iterates of our approach satisfy the following bound:

$$\mathbb{E}[\ell(\Theta^{t+1})] \leq \mathbb{E}[\ell(\Theta^t)] + \langle \nabla\ell(\Theta^t), \Theta^{t+1} - \Theta^t \rangle + \frac{L}{2} \|\Theta^{t+1} - \Theta^t\|^2. \quad (26)$$

After substituting (25) into (26), we have:

$$\begin{aligned} \mathbb{E}[\ell(\Theta^{t+1})] &\leq \mathbb{E}[\ell(\Theta^t)] - \alpha_t w_t \mathbb{E}[\|\nabla\ell(\Theta^t)\|^2] + \frac{L\alpha_t^2 w_t^2}{2} \mathbb{E}[\|\nabla\ell_{i_t}(\Theta^t)\|^2] \\ &\leq \mathbb{E}[\ell(\Theta^t)] - \alpha_t w_t \mathbb{E}[\|\nabla\ell(\Theta^t)\|^2] + \frac{L\alpha_t^2 w_t^2}{2} \sigma^2. \end{aligned} \quad (27)$$

The first inequality follows from the unbiasedness of the stochastic gradient $\mathbb{E}_{i_t}[\nabla\ell_{i_t}(\Theta^t)] = \nabla\ell(\Theta^t)$. The second inequality uses the assumption on gradient boundedness in Definition 3. Re-arranging (27) we obtain

$$\mathbb{E}[\|\nabla\ell(\Theta^t)\|^2] \leq \frac{1}{\alpha_t w_t} \mathbb{E}[\ell(\Theta^t) - \ell(\Theta^{t+1})] + \frac{L\alpha_t w_t}{2} \sigma^2. \quad (28)$$

Summing (28) from $t = 0$ to $T - 1$ and using that α_t is a fixed α , we obtain

$$\begin{aligned} \min_t \mathbb{E}[\|\nabla\ell(\Theta^t)\|^2] &\leq \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla\ell(\Theta^t)\|^2] \\ &\leq \frac{1}{T} \sum_{t=0}^{T-1} \frac{1}{\alpha w_t} \mathbb{E}[\ell(\Theta^t) - \ell(\Theta^{t+1})] + \frac{1}{T} \sum_{t=0}^{T-1} \frac{L\alpha w_t}{2} \sigma^2 \\ &\leq \frac{1}{T\alpha w_2} (\ell(\Theta^0) - \ell(\Theta^T)) + \frac{L\alpha w_1}{2} \sigma^2 \\ &\leq \frac{1}{T\alpha w_2} (\ell(\Theta^0) - \ell(\Theta^*)) + \frac{L\alpha w_1}{2} \sigma^2 \\ &\leq \frac{1}{\sqrt{T}} \left(\frac{1}{c w_2} (\ell(\Theta^0) - \ell(\Theta^*)) + \frac{L c w_1}{2} \sigma^2 \right). \end{aligned} \quad (29)$$

The first step holds because the minimum is less than the average. The second step is obtained from (28). The third step follows from the assumption on gradient weight boundedness in Definition 4. The fourth step is obtained from the fact that $\ell(\Theta^*) \leq \ell(\Theta^T)$. The final inequality follows upon using $\alpha = c/\sqrt{T}$. By setting $c = \sqrt{\frac{2(\ell(\Theta^0) - \ell(\Theta^*))}{L\sigma^2 w_1 w_2}}$ in the above inequality, we get the desired result. \square

D Proof of ℓ_{WD}

We begin with some concepts in mathematics. Let $(X, |\cdot, \cdot|)$ be a complete metric space.

Definition 5. A rectifiable curve $\gamma(t) : I \subset \mathbb{R}^+ \rightarrow X$ connecting two points p, q is called a geodesic if its length is equal to $|p, q|$ and it has unit speed. Here, we say that $\gamma(t) : I \rightarrow X$ has unit speed, if for any $s, t \in I$, $s < t$, we have, the length of the restriction

$$\gamma : [s, t] \rightarrow X$$

is $t - s$. A metric space X is called a geodesic space if, for every pair of points $p, q \in X$, there exists some geodesic connecting them.

Definition 6. We say that, a geodesic space $(X, |\cdot, \cdot|)$ has non-negative curvature in the sense of Alexandrov, if it satisfies the following property:

- for any $p \in X$, and for any unit speed geodesics $\gamma(s) : I \rightarrow X$ and $\sigma(t) : J \rightarrow X$ with $\gamma(0) = \sigma(0) := p$, the comparison angle

$$\tilde{\angle} \gamma(s) p \sigma(t) := \arccos \left(\frac{t^2 + s^2 - |\gamma(s), \sigma(t)|^2}{2 \cdot s \cdot t} \right)$$

is non-increasing with respect to each of the variables t and s .

The angle between γ and σ at p is defined by

$$\lim_{s, t \rightarrow 0^+} \arccos \left(\frac{t^2 + s^2 - |\gamma(s), \sigma(t)|^2}{2 \cdot s \cdot t} \right) \in [0, \pi].$$

In other words, every geodesic triangle in X is fatter than the one with sides length in \mathbb{R}^2 (Figure 4).

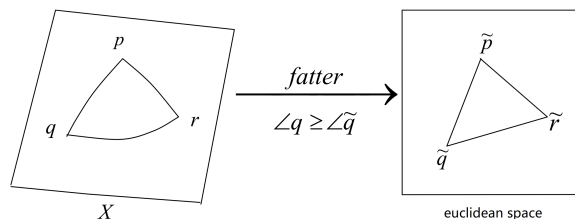


Figure 4: geodesic space with non-negative curvature

According to Sturm et al. (2006)[Proposition 2.10], the Wasserstein space $W^2(\mathbb{R}^n)$ has non-negative curvature in the sense of Alexandrov. Precisely,

Lemma 1. Sturm et al. (2006)[Proposition 2.10] Let $n \geq 1$. The Wasserstein space $W^2(\mathbb{R}^n)$ equipped with the L^2 Wasserstein distance $W_2(\cdot, \cdot)$ has non-negative curvature in the sense of Alexandrov.

Proof of Theorem 2. Let $X = W^2(\mathbb{R}^n)$ and $|\cdot, \cdot|$ be the L^2 Wasserstein distance. For any $x, y, z \in X$, we denote by γ_{xy} (γ_{zx}) the geodesic connecting x and y (resp. z and x). By the above Lemma, X has non-negative curvature in the sense of Alexandrov, hence according to Definition 6, one can define the angle between γ_{xy} and γ_{zx} at x , denoted by θ , and we have

$$\theta \geq \tilde{\angle} yxz := \arccos \left(\frac{|x, y|^2 + |z, x|^2 - |y, z|^2}{2 \cdot |x, y| \cdot |z, x|} \right),$$

which implies

$$\cos \theta \leq \frac{|x, y|^2 + |z, x|^2 - |y, z|^2}{2 \cdot |x, y| \cdot |z, x|}.$$

Equivalently,

$$|y, z|^2 \leq |x, y|^2 + |z, x|^2 - 2|x, y| \cdot |z, x| \cdot \cos \theta.$$

Hence, we complete the proof. \square

Proof of Theorem 3. We derive from the definition of ℓ_{WD} and the triangle inequality for the L^2 Wasserstein distance that for any Θ, Θ' ,

$$\begin{aligned} \|\ell_{WD}(\mathbf{u}_\Theta, \mathbf{v}) - \ell_{WD}(\mathbf{u}_{\Theta'}, \mathbf{v})\| &\leq \ell_{WD}(\mathbf{u}_{\Theta'}, \mathbf{u}_\Theta) \\ &= \ell_{WMD}^{1/2}(\mathbf{u}_{\Theta'}, \mathbf{u}_\Theta) \\ &\leq \left(\sum_{i,j} T_{i,j} d_{i,j} \right)^{1/2} \end{aligned}$$

where $T_{i,j}$ satisfies

$$\sum_j T_{i,j} = p_{u_{\Theta}, i} \quad \forall i, \quad \sum_i T_{i,j} = p_{u_{\Theta'}, j} \quad \forall j.$$

Take $T_{i,j} = \delta_{ij} \cdot p_{u_{\Theta}, i}$. According to the assumption that \mathbf{u}_Θ is Lipschitz continuous with respect to the parameters Θ , we have

$$d_{i,i} = \|u_{\Theta, i} - u_{\Theta', i}\|^2 \leq L \cdot \|\Theta' - \Theta\|^2$$

for some constant $L > 0$. Hence, we get that

$$\begin{aligned} \left(\sum_{i,j} T_{i,j} d_{i,j} \right)^{1/2} &\leq \left(\sum_i T_{i,i} \cdot L \cdot \|\Theta' - \Theta\|^2 \right)^{1/2} \\ &= \left(\sum_i T_{i,i} \right)^{1/2} \cdot L^{1/2} \cdot \|\Theta' - \Theta\| \\ &= L^{1/2} \cdot \|\Theta' - \Theta\|. \end{aligned}$$

Finally, we got

$$\|\ell_{WD}(\mathbf{u}_\Theta, \mathbf{v}) - \ell_{WD}(\mathbf{u}_{\Theta'}, \mathbf{v})\| \leq L^{1/2} \cdot \|\Theta' - \Theta\|.$$

Hence, we complete the proof. \square

E Experimental Results of Transformer

We also evaluate our method using the Transformer architecture on two translation tasks. To prevent the model from over-fitting, we use a Transformer model with a 2-layer encoder and a 2-layer decoder. Other hyper-parameters are almost the same as in Vaswani et al. (2017), except for the optimizer. In our experiment, we use SGD to train the model, instead of Adam (Vaswani et al., 2017), since our approach is derived under SGD. Results are shown in Table 3, which are consistent with the observations from the LSTM model. We hope that our approach and theoretical analysis can be extended to the Adam framework (Kingma and Ba, 2014; Chen et al., 2018b; Reddi et al., 2019) in the future.

Model	De⇒En	En⇒De	Vi⇒En	En⇒Vi
CE	29.18	24.36	25.04	26.02
CE+MLM	29.20	24.40	<u>25.68</u>	25.97
CE+BT	30.01	25.45	25.77	27.62
CE+OURS	<u>29.25</u>	<u>24.62</u>	25.49	<u>26.84</u>
WD	28.60	24.38	24.79	<u>26.43</u>
WD+MLM	<u>29.02</u>	24.49	<u>25.08</u>	26.13
WD+BT	28.92	<u>24.82</u>	24.88	26.38
WD+OURS	29.51	24.96	25.11	26.66

Table 3: BLEU scores on two translation datasets using the Transformer model. CE: Cross-Entropy loss; WD: L^2 Wasserstein distance. The best results are in **bold**, and the second-best results are in underline.