IntroVNMT: An Introspective Model for Variational Neural Machine Translation

Xin Sheng,[1] Linli Xu,[1]* Junliang Guo,[1] Jingchang Liu,[2] Ruoyu Zhao,[1] Yinlong Xu[1]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
{xins, guojunll, zry1997}@mail.ustc.edu.cn, {linlixu, ylxu}@ustc.edu.cn, jliude@cse.ust.hk

Abstract

We propose a novel introspective model for variational neural machine translation (IntroVNMT) in this paper, inspired by the recent successful application of introspective variational autoencoder (IntroVAE) in high quality image synthesis. Different from the vanilla variational NMT model, IntroVNMT is capable of improving itself introspectively by evaluating the quality of the generated target sentences according to the high-level latent variables of the real and generated target sentences. As a consequence of introspective training, the proposed model is able to discriminate between the generated and real sentences of the target language via the latent variables generated by the encoder of the model. In this way, IntroVNMT is able to generate more realistic target sentences in practice. In the meantime, IntroVNMT inherits the advantages of the variational autoencoders (VAEs), and the model training process is more stable than the generative adversarial network (GAN) based models. Experimental results on different translation tasks demonstrate that the proposed model can achieve significant improvements over the vanilla variational NMT model.

Introduction

Neural Machine Translation (NMT) has achieved remarkable success in recent years and produces superior performance over statistical machine translation (SMT). In general, most NMT models ultilize the sequence-to-sequence discriminative framework which consists of two neural networks: an RNN based encoder network transforming a source sentence $\mathbf{x} = \{x_1, x_2, \ldots, x_{Tx}\}$ into a sequence of source-side memory banks, and an RNN based decoder generating a target sentence $\mathbf{y} = \{y_1, y_2, \ldots, y_{Ty}\}$ sequentially with the use of the source-side memory banks via the attention mechanism (Bahdanau, Cho, and Bengio 2015; Luong, Pham, and Manning 2015). Due to the effectiveness and simplicity of this end-to-end model structure, NMT has been attracting more and more research interests recently. Meanwhile, despite the significant success, there still exist some challenges for standard attention based NMT models. On the one hand, traditional NMT models may not be sufficient to generate natural and accurate target sentences compared to the ground-truth, because the Maximum Likelihood Estimation (MLE) principle employed by standard NMT models is not entirely suitable for the machine translation task. More specifically, these models are only designed to maximize the posterior probability of the target ground-truth sentence given the source sentence but not to guarantee the translation quality. On the other hand, standard attention based methods cannot capture the complete information of the source sentence from the source-side memory banks (Tu et al. 2016) due to the possibility of errors in the semantic alignments between source and target words identified by the attention mechanism.

Thanks to the rise of generative models, many attempts have been made to address the challenges mentioned above with generative frameworks. To relieve the dependency of MLE, (Wu et al. 2017) adopts a generative adversarial network (GAN) based method, which directly minimizes the divergence between the distributions of the ground-truth sentences and the translations by incorporating a Convolutional Neural Network (CNN) based discriminator into the typical NMT model. To address the issues of the attention mechanism, variational NMT (VNMT) is proposed to add a latent variable into NMT which serves as a global semantic representation to facilitate generating better translations. In VNMT, in addition to the variational neural encoder and the variational neural decoder which work just like the encoder and decoder in a standard NMT model, there is a variational neural inferer that infers the latent variable **z** with a neural prior model and neural posterior approximator. Further, to alleviate the limitation of the static latent variable of VNMT, variational recurrent NMT (VRNMT) is introduced (Su et al. 2018) to incorporate a sequence of dynamic variables $\mathbf{z} = \{z_1, z_2, \ldots, z_{Ty}\}$ into the decoder input where the iteratively generated variable z_j will participate in the generation of the next target token y_j.

However, the GAN and VAE based NMT models discussed above are restricted in the following perspectives. Firstly, different from traditional GANs which assume that the space of generator is continuous, the NMT model is instead a probabilistic transformation that maps a source sentence to a target sentence which are both in discrete spaces.

[*] Corresponding author

As a consequence, the GAN based NMT model (Yu et al. 2017; Wu et al. 2017) turns to the policy gradient method named REINFORCE (Williams 1992) to build the optimization in an adversarial manner, which may lead to instability in training. Secondly, VAEs are theoretically elegant and easy to train, nevertheless VAE-based models tend to produce blurry manifold representations which are not capable of capturing sharper details, and one of the reasons could be that VAEs assign high probabilities to training samples, while not ensuring that blurry samples are assigned with low probabilities (Goodfellow, Bengio, and Courville 2016), which implies insufficient discrimination between different samples. As a consequence, the VAE based NMT models may generate some ambiguous translations which are not capable of transferring the prominent information in the source sentence in practice.

To alleviate the problems discussed above, in this paper, we propose a novel introspective model for variational neural machine translation, inspired by the recent work of introspective variational autoencoder (IntroVAE) for synthesizing high resolution images. In IntroVNMT, a different training paradigm compared to VNMT is conducted to discriminate the real target sentences from the generated ones. Specifically, the model consists of three components which play a min-max game:

- A *Variational Neural Encoder*. Just like the encoder in the vanilla NMT models which aims to encode the information of source sentences, this component transforms a source/target sentence into an intermediate representation and a sequence of memory banks.
- A *Variational Neural Inferer*. Different from the standard VNMT, this component in the proposed model self-estimates the quality of a real sentence or generated sentence. Specifically, this component attempts to assign a higher confidence score to a real target sentence and a lower score to a generated one respectively, which acts like the discriminator in GANs. To achieve that, we will reuse the two submodules in VNMT: *Neural Prior Model* and *Neural Posterior Approximator* (i.e. $p_\theta(\mathbf{z}|\mathbf{x})$ and $p_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$) which output the prior and the posterior respectively. For pairs of real source sentence $\mathbf{x}$ and real target sentence $\mathbf{y}$, this component tries to minimize the divergence between the prior and the posterior, while maximizing it for pairs of real source sentence $\mathbf{x}$ and generated target sentence $\mathbf{y}'$.
- A *Variational Neural Decoder*. Our decoder does not just work as a generator of target sentences. To complete such a min-max game, this component will focus on generating more realistic target sentences which can even mislead the Inferer. To achieve that, this component is trained with the *Variational Neural Inferer* iteratively to minimize the divergence between the prior and posterior for pairs of real source sentence $\mathbf{x}$ and generated target sentence $\mathbf{y}'$.

By training the model of IntroVNMT iteratively in an introspective manner, the three components can improve themselves and the model can eventually generate sharper target sentences. Compared to adversarial NMT models, IntroVNMT requires no extra discriminator, which reduces the complexity of the model. In the meantime, IntroVNMT can generate target sentences through a single-stream network in one stage similar to VNMT. This training paradigm integrates the advantages of the nice manifold representation of VAEs, and the adversarial training procedure of GANs, while avoiding the deficiencies of GANs and VAEs in terms of instable training and insufficient discrimination respectively.

The contribution of this work is two-fold:

- We propose the IntroVNMT model that not only self-estimates the high-level latent variables of real and generated target sentences, but also produces more realistic target sentences compared to a typical VNMT. To the best of our knowledge, this work is the first attempt to adapt IntroVAE into NMT models.
- Experimental results on the WMT'14 EN-DE and IWSLT'14 DE-EN translation tasks show that the proposed model outperforms standard VAE based NMT models by generating more realistic sentences, while distinguishing real and generated samples with significant KL-Divergence.

Related Work

Recently, end-to-end neural machine translation (NMT) has become a mainstream research direction in the field of natural language processing (NLP) (Jean et al. 2015; Bahdanau, Cho, and Bengio 2015; Wu et al. 2016; Cho et al. 2014), where many types of generative models have been proposed, including variational autoencoders (VAEs) (Kingma and Welling 2013), generative adversarial networks (GANs) (Goodfellow et al. 2014) and flow-based generative models (Kingma and Dhariwal 2018).

Neural Machine Translation. Different from statistical machine translation (SMT) which relies heavily on huge phrase/rule tables, neural machine translation (NMT) requires smaller memory. NMT starts from the sequence to sequence learning paradigm proposed by (Sutskever, Vinyals, and Le 2014), where the authors adopt two four-layer Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) models, which are responsible for encoding a source sentence into a fixed-length intermediate vector and decoding the translation step by step respectively. In order to improve the limited representation capacity of the fixed-length intermediate vector, the attention mechanism is introduced into NMT where the model can be trained to focus on the relevant parts when generating target tokens (Bahdanau, Cho, and Bengio 2015) .

Variational Autoencoders. Variational Autoencoders (VAEs) have become a group of the most prominent generative models recently. Among them, (Kingma and Welling 2013) and (Rezende, Mohamed, and Wierstra 2014) focus on VAEs which can be regarded as a regularized variant of a standard autoencoder. Specifically, VAEs introduce a neural inference model to approximate the intractable posterior probability, and optimize the model with a reparameterized variational lower bound, also called the Evidence Lower Bound (ELBO). More recently, (Huang et al. 2018) proposes a novel training paradigm for VAEs where the encoder attempts to self-estimate the latent variable and the decoder attempts to generate more realistic samples that can mislead the encoder.

Generative Adversarial Networks. Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) consist of two networks: a generator and a discriminator. In a typical adversarial procedure, the generator and discriminator are trained to compete with each other alternatively. More specifically, the generator is trained to generate more realistic samples that could fool the discriminator. Adversarial training has been applied to some natural language processing tasks successfully (Yu et al. 2017; Li et al. 2017). However, training instability is still a big challenge that GANs have to face.

VAE based Neural Machine Translation. Due to the nice manifold representation of VAEs, attempts have been made to incorporate variational latent variables into NMT training. Among them, (Bowman et al. 2016) firstly introduces a variational autoencoder to construct an unsupervised generative language model. Variational NMT (VNMT) is proposed in (Zhang et al. 2016), which introduces a latent variable produced by an extra inferer compared to a traditional encoder-decoder NMT model. VNMT is further extended in (Su et al. 2018) and (Eikema and Aziz 2019) to achieve better performance. More specifically, (Su et al. 2018) proposes variational recurrent NMT (VRNMT) models to incorporate a sequence of latent variables during decoding, while auto-encoding variational NMT (AEVNMT) is introduced in (Eikema and Aziz 2019) that generates the source and target sentences jointly from a shared latent variable. Although these VAE based NMT models can produce a manifold representation in a simple and elegant way, they are not capable of capturing sharp details in the latent variables, which restricts them from generating more realistic sentences.

The models in (Zhang et al. 2016) and (Huang et al. 2018) are most relevant to our work. In this paper, we adjust the training principle of VNMT (Zhang et al. 2016) to an adversarial manner as introduced in (Huang et al. 2018), which has been proven to be effective for generating high resolution images. Different from VRNMT (Su et al. 2018) which incorporates latent variables into hidden states of recurrent neural networks (RNNs), we concentrate on training VNMT to generate sharper manifold representations. As far as we know, this work is the first to explore the potential application of IntroVAE to NMT models.

Introspective Variational Neural Machine Translation

In this section, we introduce IntroVNMT by adapting the training paradigm of IntroVAE to vanilla VNMT. The goal is to train the model which can self-estimate the differences between the real target sentences and the generated ones, and improve itself accordingly to generate more realistic target sentences that cannot be distinguished. To achieve that, the Inferer of IntroVNMT takes the role similar to the discriminator in GANs to self-estimate the divergence between the real and generated target sentences, and maximize it for discrimination. In the meantime, the Decoder of IntroVNMT is analogous to the generator in GANs, which focuses on generating more realistic target sentences to minimize the divergence, such that it can mislead the Inferer. On the other hand, to alleviate the problems of adversarial training, the Inferer and Decoder are trained jointly, as a benefit from the manifold representations of VAEs.

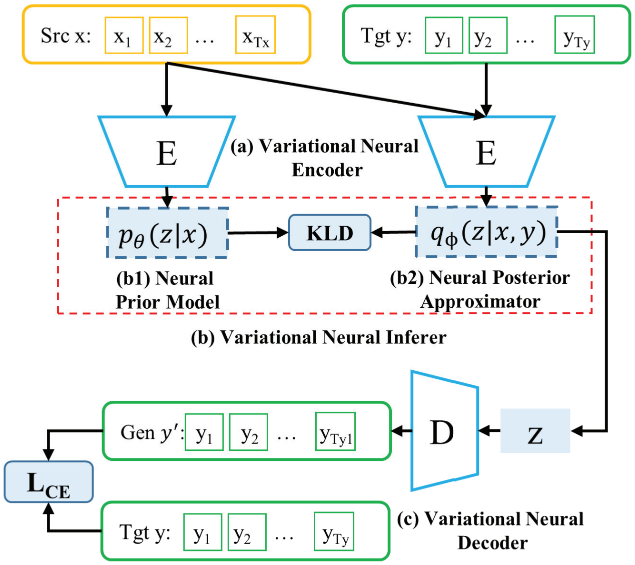

Figure 1: Framework of VNMT. 'Src' and 'Tgt' are short for 'Source' and 'Target' respectively which correspond to the ground-truth language sentences, while 'Gen' means the generated target sentence. 'E' and 'D' represent the encoder and decoder respectively. The details of the attention mechanism are not shown here for simplicity.

As shown in Fig. 1, the loss function of VNMT proposed in (Zhang et al. 2016) can be formulated in the negative version of ELBO as follows:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{y}) = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})] + \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})), \quad (1)$$

where $p_\theta(\mathbf{z}|\mathbf{x})$ is the *Neural Prior Model*, $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is the *Neural Posterior Approximator*, and $p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})$ is the *Variational Neural Decoder* conditioned on $\mathbf{z}$. Accompanied with the *Variational Neural Encoder*, VNMT conducts a single-stream training flow. More specifically, the *Variational Neural Encoder* is responsible for encoding the source/target sentences, which is the same as the encoder of NMT (Bahdanau, Cho, and Bengio 2015). The *Variational Neural Inferer* is composed of a *Neural Prior Model* and a *Neural Posterior Approximator*, where the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is encouraged to match the $p_\theta(\mathbf{z}|\mathbf{x})$ for pairs of source/target sentences. The *Variational Neural Decoder* integrates the latent representation $\mathbf{z}$ to guide the generation of target sentences (i.e. $p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})$) together with the attention mechanism (Bahdanau, Cho, and Bengio 2015).

Here we denote the two terms of ELBO as $\mathcal{L}_{CE}$ and $\mathcal{L}_{REG}$ (i.e. the cross-entropy loss and regularization loss) respectively. For the first term, we use the Monte Carlo method to approximate the expectation over the posterior,

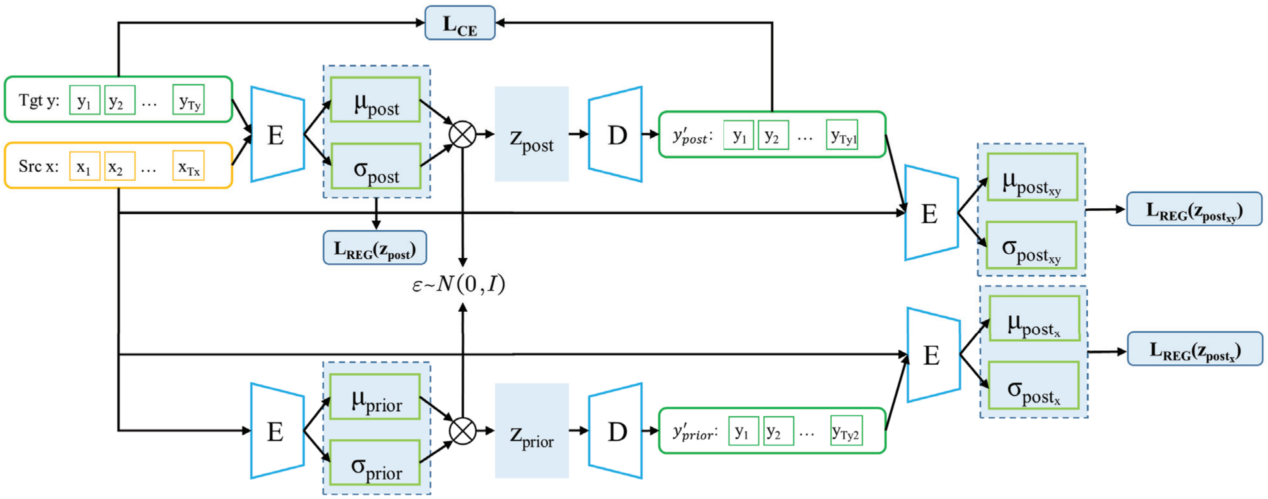

Figure 2: Framework of IntroVNMT. 'Src' and 'Tgt' are short for 'Source' and 'Target' respectively which correspond to the ground-truth language sentences. $\mathbf{y}'_{prior}$ and $\mathbf{y}'_{post}$ represent two different generated sentences. IntroVNMT consists of two components, the Inferer E and the Decoder D, in a circulation loop.

i.e. $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[*] \simeq \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{h}_z^{(l)})$, where L is the number of samples, and $\mathcal{L}_{CE}$ will degenerate to the cross-entropy loss of conventional NMTs when $L = 1$. The second term $\mathcal{L}_{REG}$ is a KL-Divergence which regularizes the encoder by encouraging the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ to match the prior $p_\theta(\mathbf{z}|\mathbf{x})$. Following the training routine described above, we will describe IntroVNMT as a modified combination of these two terms. For the sake of convenience, we integrate the *Variational Neural Encoder* and *Variational Neural Inferer* into Inferer E and keep the Decoder D intact.

Adversarial Self-Estimation

In order to self-estimate the divergence between the real and generated target sentences, we take $\mathcal{L}_{REG}$ as the adversarial training loss function. During training, the Inferer E is trained to minimize $\mathcal{L}_{REG}$ to encourage the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ of sentence pairs $(\mathbf{x},\mathbf{y})$ to match the prior $p_\theta(\mathbf{z}|\mathbf{x})$ where both $\mathbf{x}$ and $\mathbf{y}$ come from real data. Meanwhile, the Inferer E is also trained to maximize $\mathcal{L}_{REG}$ to encourage $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}')$ of sentence pairs $(\mathbf{x},\mathbf{y}')$ to deviate from the prior $p_\theta(\mathbf{z}|\mathbf{x})$, where $\mathbf{x}$ is a real source sentence while $\mathbf{y}'$ is a generated target sentence. Thus, the divergence between the distributions of real and generated target sentences will gradually increase. In contrast, the Decoder D attempts to generate target sentences that can achieve small $\mathcal{L}_{REG}$, such that the approximate posterior for the generated sentences can match the prior more closely. As a consequence, the generated target sentences will be more realistic, with a distribution closer to the real target sentences.

Given a source sentence $\mathbf{x}$, a target sentence $\mathbf{y}$ and a latent variable $\mathbf{z}$ where both $\mathbf{x}$ and $\mathbf{y}$ come from training data, we train the model as described above with the loss functions:

$$\mathcal{L}_E(\mathbf{x},\mathbf{y},\mathbf{z}) = E(\mathbf{x},\mathbf{y}) + [m - E(\mathbf{x},D(\mathbf{z}))]^+, \tag{2}$$

$$\mathcal{L}_D(\mathbf{x},\mathbf{z}) = E(\mathbf{x},D(\mathbf{z})), \tag{3}$$

where $E(\mathbf{x},\mathbf{y}) = \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x}))$, $[*]^+ = \max(0,*)$, and m is a positive margin which is set to keep $E(\mathbf{x},D(\mathbf{z}))$ not too large. Just like GANs, the Inferer E and Decoder D will play a min-max game during training.

Introspective Variational Inference

However, the training strategy introduced above may also lead to problems of model collapse and training instability, due to the characteristics of typical adversarial training. To alleviate these problems, different from the two independent models (i.e. the generator and discriminator) in GANs, we train the Inferer E and Decoder D jointly by adding a cross-entropy loss to Eq. (2) and Eq. (3). Thus, the formulation can be redefined as follows:

$$\begin{aligned} \mathcal{L}_E(\mathbf{x},\mathbf{y},\mathbf{z}) =& E(\mathbf{x},\mathbf{y}) + [m - E(\mathbf{x},D(\mathbf{z}))]^+ \\ &+ \mathcal{L}_{CE}(\mathbf{x},\mathbf{y}), \end{aligned} \tag{4}$$

$$\mathcal{L}_D(\mathbf{x},\mathbf{z}) = E(\mathbf{x},D(\mathbf{z})) + \mathcal{L}_{CE}(\mathbf{x},\mathbf{y}), \tag{5}$$

where $\mathcal{L}_{CE}(\mathbf{x},\mathbf{y})$ represents the cross-entropy loss. More specifically, this cross-entropy loss is calculated by taking the groud-truth target sentence $\mathbf{y}$ and the generated one $\mathbf{y}'$ as inputs, where $\mathbf{y}'$ is generated by taking $\mathbf{x}$ and $\mathbf{y}$ as inputs.

From the perspective of VNMT, this objective becomes the negative version of the standard VNMT's ELBO when the input pairs are $\mathbf{x}$ and $\mathbf{y}$, which preserves the nice training flow of VNMT; and from the perspective of GANs, this objective implies a min-max game between E and D when the input pairs are $\mathbf{x}$ and $\mathbf{y}'$, which facilitates the latent variables to capture sharper information.

Training of IntroVNMT

The overall training flow of IntroVNMT is shown in Fig. 2. The Inferer E is designed to output two variables μ and σ according to the input sentences, and then calculate the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ or prior $p_\theta(\mathbf{z}|\mathbf{x})$ via the reparameterization trick: $\mathbf{z} = \mu + \sigma \odot \epsilon$ where $\epsilon \sim N(0,I)$.

Algorithm 1: Training Process of IntroVNMT

```
Initialize the network parameters $\theta_D$, $\phi_E$;
while not converged do
    $X, Y \leftarrow$ Random sampled mini-batch from dataset;
    $Z_{prior} \leftarrow Enc(X)$, $Z_{post} \leftarrow Enc(X, Y)$;

    $Y'_{prior} \leftarrow D(Z_{prior})$, $Y'_{post} \leftarrow D(Z_{post})$;
    Compute the cross-entropy loss:
      $\mathcal{L}_{CE}(X, Y) \leftarrow$ cross-entropy$(Y, Y'_{post})$;
    $Z_{post_x} \leftarrow Enc(X, ng(Y'_{prior}))$,
      $Z_{post_{xy}} \leftarrow Enc(X, ng(Y'_{post}))$;
    Compute the adversarial loss for generated data:
      $\mathcal{L}^E_{adv} \leftarrow$
      $[m - \mathcal{L}_{REG}(Z_{post_x})]^+ + [m - \mathcal{L}_{REG}(Z_{post_{xy}})]^+$;
    Compute the full loss of Inferer $E$:
      $\mathcal{L}_E \leftarrow \alpha\mathcal{L}_{REG}(Z_{post}) + \beta\mathcal{L}^E_{adv} + \gamma\mathcal{L}_{CE}(X, Y)$;
    Perform gradient descent for $\phi_E$:
      $\phi_E \leftarrow \phi_E - \eta\nabla_{\phi_E}\mathcal{L}_E$;

    $Y'_{prior} \leftarrow D(Z_{prior})$, $Y'_{post} \leftarrow D(Z_{post})$;
    Compute the cross-entropy loss:
      $\mathcal{L}_{CE}(X, Y) \leftarrow$ cross-entropy$(Y, Y'_{post})$;
    $Z_{post_x} \leftarrow Enc(X, Y'_{prior})$,
      $Z_{post_{xy}} \leftarrow Enc(X, Y'_{post})$;
    Compute the adversarial loss for generated data:
      $\mathcal{L}^D_{adv} \leftarrow \mathcal{L}_{REG}(Z_{post_x}) + \mathcal{L}_{REG}(Z_{post_{xy}})$;
    Compute the full loss of Inferer $D$:
      $\mathcal{L}_D \leftarrow \beta\mathcal{L}^D_{adv} + \gamma\mathcal{L}_{CE}(X, Y)$;
    Perform gradient descent for $\phi_D$:
      $\theta_D \leftarrow \theta_D - \eta\nabla_{\theta_D}\mathcal{L}_D$;
end
```

And the input latent variable of the Decoder D is sampled from the distribution calculated by E. In this setting, the KL-Divergence $\mathcal{L}_{REG}$ (i.e. E($\mathbf{x}$, $\mathbf{y}$) in Eq. (7) and Eq. (8)) can be computed as follows:

$$
\begin{aligned}
&\mathcal{L}_{REG}(\mathbf{z}_{post}, \mathbf{z}_{prior}; \mu_{post}, \sigma_{post}, \mu_{prior}, \sigma_{prior}) \\
&= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M_z}[1 - \log(\sigma^2_{prior,ij}) + \log(\sigma^2_{post,ij}) \\
&\quad - \frac{\sigma^2_{post,ij}}{\sigma^2_{prior,ij}} - \frac{(\mu_{post,ij} - \mu_{prior,ij})^2}{\sigma^2_{prior,ij}}],
\end{aligned} \tag{6}
$$

where N is the number of samples, M_z is the dimension of the latent variable $\mathbf{z}$.

As shown in Fig. 2, the channel of generated sentences is not unique. $\mathbf{y}'_{prior}$ is generated conditioned on the latent variable $\mathbf{z}_{prior}$ which is produced only from the source sentence $\mathbf{x}$, while $\mathbf{y}'_{post}$ is generated conditioned on $\mathbf{z}_{post}$ which involves both the source sentence $\mathbf{x}$ and the target sentence $\mathbf{y}$. Thus, the full loss functions for E and D will be redefined as:

$$
\begin{aligned}
\mathcal{L}_E &= \alpha\mathcal{L}_{REG}(\mathbf{z}_{post}) + \beta\sum_{s\in S}[m - \mathcal{L}_{REG}(\mathbf{z}_s)]^+ \\
&\quad + \gamma\mathcal{L}_{CE}(\mathbf{x}, \mathbf{y}) \\
&= \alpha\mathcal{L}_{REG}(Enc(\mathbf{x}, \mathbf{y})) \\
&\quad + \beta\sum_{r\in R}[m - \mathcal{L}_{REG}(Enc(\mathbf{x}, ng(\mathbf{y}'_r)))]^+ \\
&\quad + \gamma\mathcal{L}_{CE}(\mathbf{x}, \mathbf{y}),
\end{aligned} \tag{7}
$$

$$
\begin{aligned}
\mathcal{L}_D &= \beta\sum_{s\in S}\mathcal{L}_{REG}(\mathbf{z}_s) + \gamma\mathcal{L}_{CE}(\mathbf{x}, \mathbf{y}) \\
&= \beta\sum_{r\in R}\mathcal{L}_{REG}(Enc(\mathbf{x}, \mathbf{y}'_r)) + \gamma\mathcal{L}_{CE}(\mathbf{x}, \mathbf{y}),
\end{aligned} \tag{8}
$$

where $S = \{post_{xy}, post_x\}$, $R = \{post, prior\}$, $\mathcal{L}_{REG}(\mathbf{z})$ denotes $\mathcal{L}_{REG}(\mathbf{z}, \mathbf{z}_{prior})$ for simplicity, Enc represents the Inferer E, $\mathcal{L}_{CE}(\mathbf{x}, \mathbf{y})$ is the cross-entropy loss in the standard VNMT loss which takes $\mathbf{x}$ and $\mathbf{y}$ as input, $ng(*)$ indicates that the update of the gradients is stopped, α, β and γ are three hyper-parameters to balance the different parts of the loss function.

The complete algorithm is summarized in Algorithm 1, where the Inferer E and Decoder D play a min-max game just like GANs. More specifically, the Inferer E and Decoder D are trained iteratively, such that E is updated to distinguish the three different types of sentence pairs (i.e. pairs of real source/target sentence $X\&Y$ and pairs of real source sentence and generated target sentence $X\&Y'_{prior}$ or $X\&Y'_{post}$), while D is updated to generate target sentences that are increasingly similar to real ones. It is worth noting that directly providing discrete generated target sentences as input to the Inferer E does not allow for backpropagation as they are discontinuous. Here we use the straight-through Gumbel-Softmax approximation (Jang, Gu, and Poole 2016) at the output of the Decoder D to sample words during training.

Experiments

In this section, we conduct experiments on the WMT'14 English→German (EN-DE for short) and IWSLT'14 German→English (DE-EN for short) translation tasks to demonstrate the effectiveness of IntroVNMT.

Data Settings

For the EN-DE translation task, we use the same datasets as (Zhang et al. 2016). Our training set[1] consists of 4.45M sentence pairs with 116.1M English words and 108.9M German words. We use news-test 2013 as the validation set and news-test 2015 as the test set. For the DE-EN translation task, we select the dataset from the IWSLT 2014 evaluation campaign (Cettolo et al. 2014), consisting of training/validation/test corpus with approximately 153K, 7K and 6.5K bilingual sentence pairs respectively.

[1] The preprocessed data can be found and downloaded from http://nlp.stanford.edu/projects/nmt/

Table 1: Case-sensitive BLEU scores of different NMT systems on the WMT'14 EN-DE translation task. Here we use the case-sensitive BLEU scores as the evaluation metric. The default setting of various NMT models follows conventional RNNSearch (Bahdanau, Cho, and Bengio 2015).

System	**System Structure**	**BLEU**
RNNSearch related NMT models		
RNNSearch (Bahdanau, Cho, and Bengio 2015)	Bidirectional encoder + Word-level decoder	23.4
BPEChar (Chung, Cho, and Bengio 2016)	Bidirectional encoder + Character-level decoder	23.9
RecAtten (Yang et al. 2017)	RNNSearch + Recurrent attention	25.0
ConvEncoder (Gehring et al. 2017)	Convolutional encoder + Word-level decoder	24.3
VAE based NMT models		
VNMT (Zhang et al. 2016)	RNNSearch + Static latent variable	25.49
VRNMT (Su et al. 2018)	RNNSearch + Dynamic latent variable	25.93
IntroVNMT (this work)	RNNSearch + Static latent variable + Introspective training	**26.14**

Table 2: Case-insensitive BLEU scores of different systems on the IWSLT'14 DE-EN translation task. Here we conduct experiments only for **RNNSearch**, **VNMT**, **VRNMT** and the proposed **IntroVNMT**.

System	**BLEU**
RNNSearch	28.36
VNMT	28.77
VRNMT	29.39
IntroVNMT (this work)	**29.7**

To improve the computational efficiency and avoid problems with closed vocabularies, we segment the data using byte pairs encoding (BPE) (Sennrich, Haddow, and Birch 2016), except for the target language sentences in the test set which are left as the ground-truth for testing. Sentences longer than 50 words are removed. For the EN-DE task, we take the vocabularies produced by BPE as our final vocabularies. For the DE-EN task, the vocabularies for the German and English corpus include about 23K and 32K most frequent words respectively. All the other words not in the vocabulary are replaced with a special token 'UNK'. Finally, we use BLEU (Papineni et al. 2002) as our evaluation metric. Here we emloy the case-sensitive BLEU score for the EN-DE translation task, and the case-insensitive BLEU score for the DE-EN translation task.

Baseline Methods

We compare our model against the following systems:

RNNSearch. For convenient comparison with previous VAE based NMT models, we keep the basic model as RNNSearch (Bahdanau, Cho, and Bengio 2015), which is an RNN-based encoder-decoder framework.

VNMT. It is a variational NMT model proposed in (Zhang et al. 2016) that incorporates a static latent variable into the model, which serves as a global semantic signal for the sentence pairs.

VRNMT. Different from VNMT, it introduces dynamic latent variables instead of a static latent variable to model the translation procedure of a sentence.

Training Details

The implementation of various NMT models is based on RNNSearch (Bahdanau, Cho, and Bengio 2015), so we follow the settings of RNNSearch, except for some hyper-parameters specific to different NMT models. For the basic hyper-parameters, we set the word embedding dimension as 620, hidden layer size as 1000, learning rate as 1×10^{-4}, batch size as 80, gradient norm as 1.0 and dropout rate as 0.3 (Srivastava et al. 2014). As implemented in the VAE framework, we set the sampling number $L = 1$ and the dimension of the latent variable as 2000. During decoding, we adopt the beam search algorithm (Sutskever, Vinyals, and Le 2014) and set the beam size as 10 for all models. For IntroVNMT, we set $m = 100, \alpha = 1, \beta = 1$ and $\gamma = 1$ as the default parameters respectively. The Inferer E and Decoder D are trained iteratively using the Adam algorithm (Kingma and Ba 2015) ($\beta_1 = 0.9, \beta_2 = 0.999$).

We initialize the parameters of various VAE based models and IntroVNMT with the pretrained RNNSearch model and the pretrained VNMT model respectively. With regard to the source and target encoders, we share the parameters of GRUs except for the word embeddings.

Results on WMT'14 EN-DE Translation

In Table 1, we compare the performance of IntroVNMT with several previous works (Bahdanau, Cho, and Bengio 2015; Chung, Cho, and Bengio 2016; Yang et al. 2017; Gehring et al. 2017; Zhang et al. 2016; Su et al. 2018). Here we directly report the results of **BPEChar**, **RecAtten** and **ConvEncoder** as provided in (Gehring et al. 2017) and the results of VAE based NMT models as provided in (Su et al. 2018). As shown in Table 1, IntroVNMT outperforms the two previous VAE based NMT models including VNMT and VRNMT with gains of 0.65 and 0.21 BLEU points respectively. Meanwhile, our model also achieves comparable performance to several recent NMT models.

Results on IWSLT'14 DE-EN Translation

Here we compare the VAE based models on the IWSLT'14 DE-EN Translation task to verify the improvements of introspective training with the measure of case-insensitive BLEU

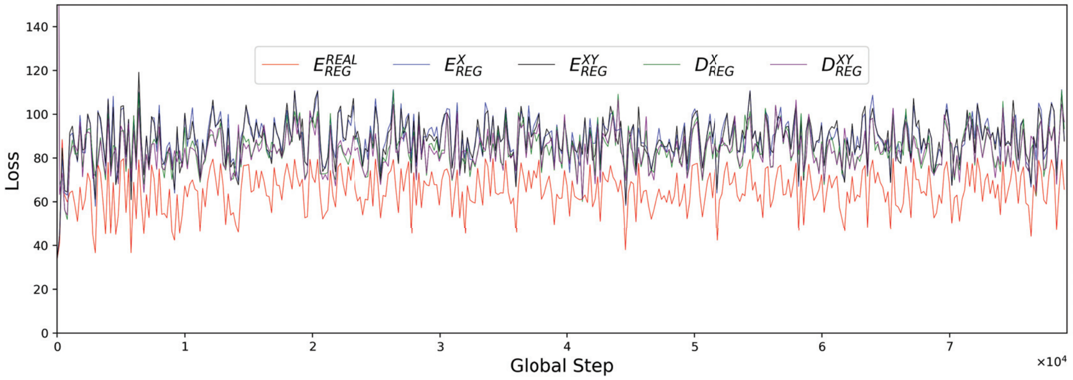

Figure 3: The training process of IntroVNMT on DE-EN translation. Here we plot the curves of different KL-Divergence losses during training. Specifically, E^{REAL}_{REG}, E^{X}_{REG}, E^{XY}_{REG} represent $\mathcal{L}_{REG}(\mathbf{z}_{post})$, $\mathcal{L}_{REG}(\mathbf{z}_{post_x})$, $\mathcal{L}_{REG}(\mathbf{z}_{post_{xy}})$ for the Inferer E. D^{X}_{REG}, D^{XY}_{REG} correspond to $\mathcal{L}_{REG}(\mathbf{z}_{post_x})$, $\mathcal{L}_{REG}(\mathbf{z}_{post_{xy}})$ for the Decoder D.

Table 3: Translation results generated by different NMT systems on the DE-EN task. The important parts are in bold.

Source	*genau dies ist der grund, warum wir musik machen: damit wir etwas,* *das in uns allen, tief im inneren steckt, unsere gefühle,* *durch unsere künstlerische linse,* ***durch unsere kreativität zur wirklichkeit formen können**.*	*sofort bekommen wir einen eindruck über die themen,* ***die auf wikipedia am populärsten sind**.*
Ground-Truth	*this was the very reason why we made music,* *that we take something that exists within all of* *us at our very fundamental core,* *our emotions, and through our artistic lens,* ***through our creativity,*** ***we're able to shape those emotions into reality**.*	*right away, we get a sense of what are the topical domains* ***that are most popular on wikipedia**.*
VNMT	*this is the reason why we make music:* *that we put something that in all of us,* *is inside us, deep inside, our feelings,* *our artistic lens, through our artistic lens,* *can form through our artistic lens* ***through our creativity**.*	*now we get a sense of the issues* ***who are at wikipedia at wikipedia**.*
VRNMT	*this is the reason why we do music:* *so we're doing something that is in our midst,* *deep inside, our emotions, through our artistic lens,* ***through our creativity**.*	*and then we get a sense of the topics* ***that are on wikipedia**.*
IntroVNMT	*so this is why we make music:* *so we have something that's in all of us, in the interior,* *our feelings, through our artistic lens,* ***through our creativity to reality**.*	*immediately , we get a sense of the issues* ***that are on wikipedia at the most popular**.*

scores. As shown in Table 2, IntroVNMT model significantly enhances the translation quality and achieves gains of 0.93 and 0.31 over VNMT and VRNMT respectively, indicating that the incorporation of introspective training is effective for improving variational NMT. Moreover, by comparing the BLEU scores of VNMT and IntroVNMT, we can find that the introspective training principle can significantly improve the translation performance.

Analysis of Training Stability

To illustrate the training process of IntroVNMT, in Fig. 3 we plot the curves of different KL-Divergence losses which converge quickly to stable states with values fluctuating steadily, justifying the effectiveness of the nice manifold representation for stable training. From Fig. 3, we have two observations: 1) All the values of E^{X}_{REG}, E^{XY}_{REG}, D^{X}_{REG} and D^{XY}_{REG} are very close, since the origin of the two different types of generated sentences is the same. Specifically, the information of $\mathbf{z}_{post}$ and $\mathbf{z}_{prior}$ both come from real source/target sentences which express the same meaning; 2) The distinction between E^{REAL}_{REG} and the other four is significant, which indicates that the Inferer can self-estimate the quality of the real and generated target sentences clearly.

Analysis of High Quality Translation Results

We further compare the quality of translations produced by different VAE based NMT models with examples as shown in Table 3. Clearly, VRNMT outperforms VNMT in terms of over-translation in both cases, which has been demonstrated in (Su et al. 2018). However, among all the models, only IntroVNMT can highlight the skeleton of a source sentence in its translation. As shown in the left example, the full meaning of "*durch unsere kreativität zur wirklichkeit formen können*" in the source language is expected to be translated to "*through our creativity, we're able to shape those emotions into reality*" in the target language. But only IntroVNMT expresses it as "*through our creativity to reality*", which verifies that IntroVNMT can generate sharper latent variables. Similarly, only IntroVNMT can express the most prominent information "most popular" in the right shorter example.

Conclusion

This paper presents an IntroVNMT model that introduces an introspective training paradigm for variational neural machine translation. Similar to GANs, the learning objective of IntroVNMT plays a min-max game between the Inferer and Decoder. The Inferer E is trained to not only self-estimate the quality of real and generated target sentences, but also produce nice manifold representation. The Decoder D is trained to generate more realistic target sentences. These two parts are trained iteratively and improve themselves accordingly. Compared to VNMT and VRNMT, our model can capture the skeleton information of source sentences while maintaining the simple framework of VNMT. It is worth noting that IntroVNMT is extended from standard VNMT models, therefore it is orthogonal to some state-of-the-art methods, and can be applied to them (such as Transformer) for further improvements.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (No. 61673364, No. 61772486) and the Fundamental Research Funds for the Central Universities (WK2150110008). The authors would also like to thank the Information Science Laboratory Center of USTC for the hardware & software services.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In *Proc. of ICLR*.

Cettolo, M.; Niehues, J.; Stüker, S.; Bentivogli, L.; and Federico, M. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, 57.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proc. of ACL*.

Chung, J.; Cho, K.; and Bengio, Y. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proc. of ACL*.

Eikema, B., and Aziz, W. 2019. Auto-encoding variational neural machine translation. In *Proc. of ACL*.

Gehring, J.; Auli, M.; Grangier, D.; and Dauphin, Y. N. 2017. A convolutional encoder model for neural machine translation. In *Proc. of ACL*.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proc. of NIPS*, 2672–2680.

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

Huang, H.; He, R.; Sun, Z.; Tan, T.; et al. 2018. Introvae: Introspective variational autoencoders for photographic image synthesis. In *Proc. of NIPS*, 52–63.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2015. On using very large target vocabulary for neural machine translation. In *Proc. of ACL*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Kingma, D. P., and Dhariwal, P. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Proc. of NIPS*, 10215–10224.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. In *Proc. of ACL*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proc. of ACL*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, 311–318.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. of ICML*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *Proc. of ACL*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15(1):1929–1958.

Su, J.; Wu, S.; Xiong, D.; Lu, Y.; Han, X.; and Zhang, B. 2018. Variational recurrent neural machine translation. In *Proc. of AAAI*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*, 3104–3112.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016. Modeling coverage for neural machine translation. In *Proc. of ACL*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wu, L.; Xia, Y.; Zhao, L.; Tian, F.; Qin, T.; Lai, J.; and Liu, T.-Y. 2017. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*.

Yang, Z.; Hu, Z.; Deng, Y.; Dyer, C.; and Smola, A. 2017. Neural machine translation with recurrent attention modeling. In *Proc. of EACL*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proc. of AAAI*.

Zhang, B.; Xiong, D.; Su, J.; Duan, H.; and Zhang, M. 2016. Variational neural machine translation. In *Proc. of AAAI*.