TILGAN: Transformer-based Implicit Latent GAN for Diverse and Coherent Text Generation

Shizhe Diao[♦*], **Xinwei Shen**[♦*], **KaShun Shum**[♦], **Yan Song**[♠♡], **Tong Zhang**[♦]
[♦]The Hong Kong University of Science and Technology
{sdiaoaa, xshenal, ksshumab, tongzhang}@ust.hk
[♠]The Chinese University of Hong Kong (Shenzhen)
[♡]Shenzhen Research Institute of Big Data
songyan@cuhk.edu.cn

Abstract

Conventional autoregressive models have achieved great success in text generation but suffer from the exposure bias problem in that token sequences in the training and in the generation stages are mismatched. While generative adversarial networks (GANs) can remedy this problem, existing implementations of GANs directly on discrete outputs tend to be unstable and lack diversity. In this work, we propose **TILGAN**, a **T**ransformer-based **I**mplicit **L**atent **GAN**, which combines a Transformer autoencoder and GAN in the latent space with a novel design and distribution matching based on the Kullback-Leibler (KL) divergence. Specifically, to improve local and global coherence, we explicitly introduce a multi-scale discriminator to capture the semantic information at varying scales among the sequence of hidden representations encoded by Transformer. Moreover, the decoder is enhanced by an additional KL loss to be consistent with the latent-generator. Experimental results on three benchmark datasets demonstrate the validity and effectiveness of our model, by obtaining significant improvements and a better quality-diversity trade-off in automatic and human evaluation for both unconditional and conditional generation tasks.[1]

1 Introduction

In recent years, Transformer-based autoregressive (AR) models have made a dramatic impact in text generation tasks such as machine translation (Vaswani et al., 2017; Wang et al., 2019) and dialogue systems (Le et al., 2019; Ham et al., 2020), especially with the emergence of large pre-trained language models (Radford et al., 2019; Brown et al., 2020; Wu et al., 2020). However, AR models predict the next token conditioned on the ground truth during training and on its own previously generated token during inference, which leads to a mismatch between training and generation stages, and this causes low quality of generated texts and bad generalization ability of models on unseen data (Wiseman and Rush, 2016; Welleck et al., 2020).

Generative adversarial networks (GANs, Goodfellow et al., 2014) provide a promising approach to solve the exposure bias problem (Yu et al., 2017; Kusner and Hernández-Lobato, 2016; Zhang et al., 2017). This is because GANs aim at matching the distributions of the generated and real data instead of forcing the model output to align with the single correct sequence, and thus provide the potential to bypass the discrepancy issue. However, it is non-trivial to apply GANs to discrete data since the gradients cannot be normally back-propagated through discrete tokens. Existing approaches have implemented the adversarial discrete generation training by reinforcement learning (RL) (Yu et al., 2017; Lin et al., 2017; Guo et al., 2018; Fedus et al., 2018) and Gumbel-Softmax (Kusner and Hernández-Lobato, 2016). Nevertheless, these approaches suffer from the high variance problem which causes unstable performance and slow convergence, leading to other methods based on feature matching (Zhang et al., 2017; Zhao et al., 2018; Chen et al., 2018).

In this work, we propose **TILGAN**, a **T**ransformer-based **I**mplicit **L**atent **GAN**, which combines a Transformer autoencoder and a GAN in the latent space with novel designs and a learning formulation based on the Kullback-Leibler (KL) divergence to enhance the text generation performance in both fidelity and diversity. Specifically, inspired by the representation capacity of Transformer AR models, we firstly incorporate Transformer architectures to improve GANs in text generation. Note that the previous latent feature matching methods are mostly RNN-based and assume

[*]Equal Contribution.
[1]Our code is available at https://github.com/shizhediao/TILGAN.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4844–4858

a single vector in the latent space, which do not directly handle a sequence of latent representations encoded by a Transformer. However, single latent vector representation hinders the incorporation of correlations among different tokens, leading to the loss of crucial semantic information captured by a Transformer structure. This is especially problematic for local and global coherence (Bińkowski et al., 2020). In this paper, we directly match the distributions of multi-token sequences in the latent space, which is better suited for the Transformer structure. To do so, we have to resolve two challenges, the first being how to do distribution matching. We introduce a multi-scale discriminator over the Transformer latent space to utilize the semantic information on different scales, where a global discriminator takes the entire sequence of latent representations as the input, and a local discriminator takes only a randomly-sampled local neighborhood. The second challenge is how to train the decoder reliably. We enhance an autoencoder loss by another KL loss optimized by GAN, forcing the latent representations of the decoding output to be compatible with the generated latent representations from the latent-generator.

We provide a theoretical justification for the proposed formulation by connecting it to the standard goal of generative modeling. Experimental results on three datasets illustrate that TILGAN outperforms all baselines in both unconditional and conditional generation tasks, achieving state-of-the-art performance. Particularly, TILGAN exhibits a better quality-diversity trade-off evaluated by automatic metrics such as SelfBLEU and TestBLEU as well as human evaluation. Further analyses also confirm the effectiveness of each component of our method, where decoder enhancement greatly benefits generation quality, while the multi-scale discriminator and KL objective provide great performance gains in generation diversity, and the implicit prior contributes to both.

2 The Approach

2.1 Model and Formulation

In this section, we introduce the proposed model and the learning formulation. Let $\boldsymbol{x} \in \mathcal{X}$ denote a sentence following the real data distribution $p_r(\boldsymbol{x})$ with $\mathcal{X} = \mathcal{V}^n$ where $\mathcal{V}$ is the vocabulary, $m = |\mathcal{V}|$ is the vocabulary size, and n is the sequence length, and $\boldsymbol{z} \in \mathcal{Z}$ be the latent variable following a prior distribution $p_z(\boldsymbol{z})$. We consider a probabilistic model containing an encoder $E_\phi : \mathcal{X} \to \mathcal{Z}$ and a decoder $G_\theta : \mathcal{Z} \to \mathcal{X}$. Both are generally stochastic mappings with parameters ϕ and θ, and induce the encoder conditional distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the decoder conditional $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ respectively. Note that previous approaches to text generation use deterministic encoders and decoders (Zhao et al., 2018), which restricts the expressiveness of the modeled distribution family. We first ensure the consistency between E_ϕ and G_θ by minimizing the negation of the expected reconstruction log-likelihood

$$L_c(\phi, \theta) = -\mathbb{E}_{\boldsymbol{x}\sim p_r}\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})], \quad (1)$$

which coincides with the reconstruction term in the evidence lower bound (ELBO).

The generated data distribution is given by $p_G(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim p_z}[p_\theta(\boldsymbol{x}|\boldsymbol{z})]$. To achieve good generation performance, we design the model so that the distribution family of $p_G(\boldsymbol{x})$ is large enough to contain the real one $p_r(\boldsymbol{x})$. As described in Section 2.3, we use Transformer to model E and G, which we assume to have sufficient capacity to reconstruct data well and learn informative latent representations. In this way, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is assumed to be expressive enough. To further enhance the capacity of $p_G(\boldsymbol{x})$, we propose to use an implicit prior p_z, by transforming samples from a simple distribution with a deep neural network.

Consider a random vector $\epsilon \in \mathcal{E}$ following some simple distribution p_ϵ like a standard Gaussian. We then propose to learn a latent-generator $g_\beta : \mathcal{E} \to \mathcal{Z}$ with parameter β so that the distribution of $g_\beta(\epsilon)$ matches that of $E_\phi(\boldsymbol{x})$, by minimizing the KL divergence

$$L_g(\phi, \beta) = D_{\mathrm{KL}}(q_\phi(\boldsymbol{z})\|p_\beta(\boldsymbol{z})),$$

where $p_\beta(\boldsymbol{z})$ denotes the distribution of $g_\beta(\epsilon)$ and $q_\phi(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x}\sim p_r}[q_\phi(\boldsymbol{z}|\boldsymbol{x})]$ is the distribution of $E(\boldsymbol{x})$, a.k.a., the aggregated posterior. The advantage of KL divergence is that it imposes a heavy penalty when $q_\phi(\boldsymbol{z}) > 0$ but $p_\beta(\boldsymbol{z}) \approx 0$, which means that it favors a g that covers all the diverse modes of $q_\phi(\boldsymbol{z})$. This is commonly known and verified empirically in Shen et al. (2020). Hence minimizing KL encourages a better diversity in generation compared with the Jensen–Shannon (JS) divergence or Wasserstein distance which are often used in the literature on generative models.

Therefore, we formulate the overall objective

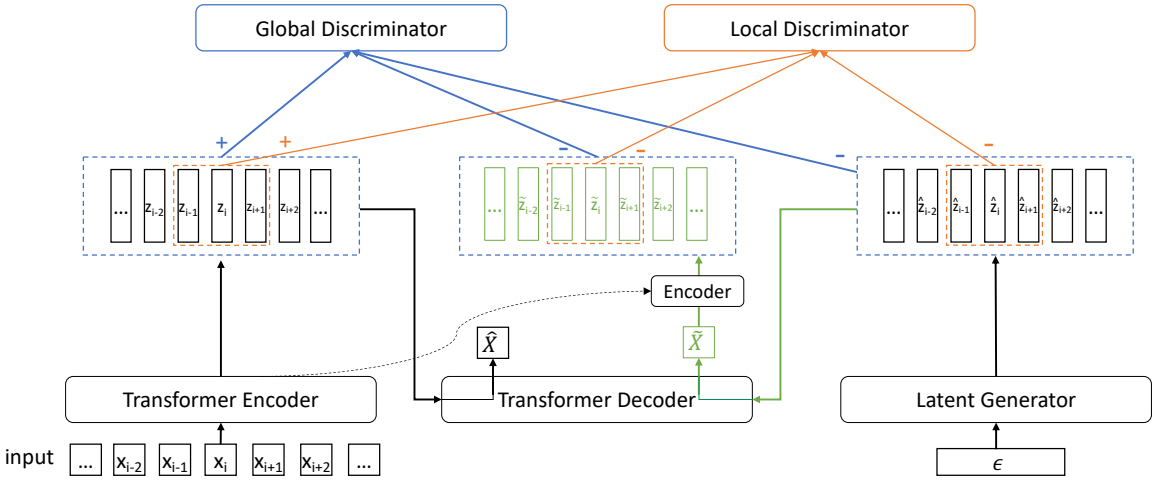

Figure 1: The overall architecture of TILGAN. Blue and orange stand for the global and local discriminators, respectively, and green denotes the route of the enhanced decoder.

function to be minimized as follows

$$L_c(\phi, \theta) + \lambda L_g(\phi, \beta), \tag{2}$$

where $\lambda > 0$ is a coefficient to balance both terms.

Decoder Enhancement During testing, a new sentence is generated by first sampling $\epsilon \sim p_\epsilon$, then computing the latent variable $g(\epsilon)$ and finally generating $G(g(\epsilon))$, which means the decoder takes the output of the latent-generator g as the input which it has never seen throughout the training. Although the KL term aims at matching the distributions of $g(\epsilon)$ and $E(\boldsymbol{x})$, it is possible that they do not match perfectly. In such cases, the decoder may generate data with poor fidelity and far from being real data. To resolve this and reliably train the decoder, we propose to enhance the decoder by letting it see the generated latent $g(\epsilon)$ during training. Formally, let $\tilde{p}_g$ be the distribution of $E(G(g(\epsilon)))$. We add another term to the loss function (2) with coefficient $\lambda_1 > 0$:

$$\lambda_1 D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}) \| \tilde{p}_g(\boldsymbol{z})). \tag{3}$$

Since this term is designed to enhance the decoder, we regard the encoder and prior parameters ϕ and β as fixed constants. In other words, in optimization, we do not propagate gradients of this term with respect to ϕ and β and only update the decoder parameter θ.

2.2 Algorithm

In this section, we propose a GAN-based algorithm for the optimization of the above formulation. Since $p_\beta(\boldsymbol{z})$ is implicit, the KL term L_g in (2) does not allow a closed form to be optimized directly. We introduce a discriminator to estimate the gradients, following Shen et al. (2020). In Lemma 1, we present the gradient formulas of L_g.

Lemma 1. *Let* $\mathcal{D}(\boldsymbol{z}) = \ln(q_\phi(\boldsymbol{z})/p_\beta(\boldsymbol{z}))$. *Then*

$$\nabla_\phi L_g = \mathbb{E}[\nabla_{\boldsymbol{z}} \mathcal{D}(E_\phi(\boldsymbol{x}))^\top \nabla_\phi E_\phi(\boldsymbol{x})],$$
$$\nabla_\beta L_g = -\mathbb{E}[s_{\mathcal{D}}(g_\beta(\epsilon)) \nabla_{\boldsymbol{z}} \mathcal{D}(g_\beta(\epsilon))^\top \nabla_\beta g_\beta(\epsilon)],$$

where $s_{\mathcal{D}}(\boldsymbol{z}) = e^{\mathcal{D}(\boldsymbol{z})}$ *is the scaling factor and the expectations are taken over all the randomness.*

Since $\mathcal{D}$ depends on the unknown densities q_ϕ and p_β so that the gradients in Lemma 1 can not be directly computed from the data, we estimate the gradients by training a discriminator D_ψ with parameter ψ via the empirical logistic regression:

$$\min_\psi \left[\sum_{\boldsymbol{z} \in S_e} \frac{\ln(1 + e^{-D_\psi(\boldsymbol{z})})}{|S_e|} + \sum_{\boldsymbol{z} \in S_g} \frac{\ln(1 + e^{D_\psi(\boldsymbol{z})})}{|S_g|} \right],$$

where S_e and S_g are finite samples from $q_\phi(\boldsymbol{z})$ and $p_\beta(\boldsymbol{z})$ respectively. This leads to a GAN algorithm. The optimization of the enhanced loss (3) is similar.

However, GAN is commonly known to suffer from unstable training or gradient vanishing. To stabilize our algorithm, we adopt the scaling clipping technique from Shen et al. (2020) and clip the scaling factor into a range of $[r_0, 1/r_0]$, where $r_0 = 0.5$ turns out to work well in all our experiments. Denote the clipped scaling by $s'_D(\boldsymbol{z}) = \max\{\min\{s_D(\boldsymbol{z}), 2\}, 0.5\}$.

For the optimization of the consistency loss L_c, we adopt the reparametrization trick from Kingma and Welling (2014) and estimate it by

Algorithm 1: TILGAN

Input: initial $\phi, \theta, \beta, \psi, \xi$, batch-size N, local size M
while *not convergence* **do**
  // Update discriminators
  Sample $\{\boldsymbol{x}_i\}_{i=1}^N \sim p_r(\boldsymbol{x})$, $\{\epsilon_i\}_{i=1}^N \sim p_\epsilon$
  Compute
  $\boldsymbol{z}_i = E_\phi(\boldsymbol{x}_i), \hat{\boldsymbol{z}}_i = g_\beta(\epsilon_i), \tilde{\boldsymbol{z}}_i = E_\phi(G_\theta(\hat{\boldsymbol{z}}_i))$
  Update ψ by descending the gradient:
  $\frac{1}{N}\sum_{i=1}^N \nabla_\psi[\ln(1+e^{-D_\psi(\boldsymbol{z}_i)})+\ln(1+e^{D_\psi(\hat{\boldsymbol{z}}_i)})]$
  Randomly sample local blocks $\boldsymbol{z}'_i$ and $\hat{\boldsymbol{z}}'_i$ with size M
  Update local discriminator by descending:
  $\frac{1}{N}\sum_{i=1}^N \nabla_\xi[\ln(1+e^{-d_\xi(\boldsymbol{z}'_i)})+\ln(1+e^{d_\xi(\hat{\boldsymbol{z}}'_i)})]$
  // Update encoder, decoder and latent-generator
  Obtain $\boldsymbol{x}_i, \epsilon_i, \boldsymbol{z}_i, \hat{\boldsymbol{z}}_i, \tilde{\boldsymbol{z}}_i, \boldsymbol{z}'_i$ and $\hat{\boldsymbol{z}}'_i$ as above
  Compute ϕ-gradient:
  $\frac{1}{N}\sum_{i=1}^N[\nabla_\phi \hat{L}_c(\boldsymbol{x}_i, \boldsymbol{z}_i)+\lambda\nabla_\phi D_\psi(\boldsymbol{z}_i)+\lambda\nabla_\phi d_\xi(\boldsymbol{z}'_i)]$
  Compute β-gradient:
  $-\frac{1}{N}\sum_{i=1}^N \lambda[s'_D(\hat{\boldsymbol{z}}_i)\nabla_\phi D_\psi(\hat{\boldsymbol{z}}_i)+s'_d(\hat{\boldsymbol{z}}'_i)\nabla_\phi d_\xi(\hat{\boldsymbol{z}}'_i)]$
  Compute θ-gradient:
  $\frac{1}{N}\sum_{i=1}^N[\nabla_\theta \hat{L}_c(\boldsymbol{x}_i, \boldsymbol{z}_i)+\lambda_1 s'_D(\tilde{\boldsymbol{z}}_i)\nabla_\phi D_\psi(\tilde{\boldsymbol{z}}_i)]$
  Update parameters ϕ, θ, β using the gradients
Return: ϕ, θ, β

$\frac{1}{n}\sum_{i=1}^n \hat{L}_c(\boldsymbol{x}_i, \boldsymbol{z}_i)$ where $\boldsymbol{x}_i \sim p_r(\boldsymbol{x})$, $\boldsymbol{z}_i = E_\phi(\boldsymbol{x}_i)$, and $\hat{L}_c(\boldsymbol{x}_i, \boldsymbol{z}_i) = -\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)$. The whole training procedure is summarized in Algorithm 1, where the colored parts stand for the enhanced decoder (green) and the multi-scale discriminator (blue and orange) introduced later.

2.3 Architecture

In this section, we present the Transformer-based architecture incorporated with multi-scale discriminators. We propose a Transformer autoencoder framework where both the encoder and decoder are self-attention layers with three novel ingredients specific to improve the generation performance in both quality and diversity: (i) a latent-generator g to transform Gaussian noises into an implicit prior distribution, (ii) decoder enhancement, and (iii) multi-scale discriminators. Figure 1 illustrates the entire architecture of TILGAN.

As mentioned in Section 1, we introduce multiple discriminators over the Transformer's latent space to utilize the semantic information on different scales, each of which operates on a different window of representations as the input. Specifically, given an input sentence $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ where x_i stands for the i-th word, it is passed through the Transformer encoder which results in a sequence of latent states $\boldsymbol{z} = [z_1, z_2, \ldots, z_n]$ where z_i is the vector representation corresponding to x_i. We introduce a global discriminator D_ψ taking the whole sequence of representations $\boldsymbol{z}$ as the input, and a local discriminator d_ξ with parameter ξ taking only a local neighborhood of the M randomly-sampled adjacent representations, e.g., $\boldsymbol{z}' = [z_{i-1}, z_i, z_{i+1}]$ with $M = 3$, as the input. [2] Notably, the local discriminator takes the generated pieces of sequences into account, so it provides signals of phrase-level fidelity and local coherence, while the global discriminator is able to assess the general realism and the degree of natural coherence for the whole sequence.

2.4 Extension to Conditional Generation

Our proposed framework can be readily extended to conditional generation tasks such as story completion. To be specific, the goal is to learn a conditional real data distribution $p_r(\boldsymbol{x}|\boldsymbol{c})$ where $\boldsymbol{c}$ is the given context following $p_r(\boldsymbol{c})$ with some missing content $\boldsymbol{x}$ to complete. We propose to feed $\boldsymbol{c}$ into all three components—encoder E, decoder G, and latent-generator g—of our model, and modify the terms in objective function (2) as follows

$$L'_c(\phi, \theta) = -\mathbb{E}_{p_r(\boldsymbol{x},\boldsymbol{c})}\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})],$$
$$L'_g(\phi, \beta) = D_{\mathrm{KL}}(q^{\boldsymbol{c}}_\phi(\boldsymbol{z})\|p^{\boldsymbol{c}}_\beta(\boldsymbol{z})),$$

where $p_r(\boldsymbol{x}, \boldsymbol{c}) = p_r(\boldsymbol{x}|\boldsymbol{c})p_r(\boldsymbol{c})$, and the marginal distributions of $E(\boldsymbol{x}, \boldsymbol{c})$ and $g(\epsilon, \boldsymbol{c})$ are given by $q^{\boldsymbol{c}}_\phi(\boldsymbol{z}) = \mathbb{E}_{p_r(\boldsymbol{x},\boldsymbol{c})}[q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})]$ and $p^{\boldsymbol{c}}_\beta(\boldsymbol{z}) = \mathbb{E}_{p_r(\boldsymbol{x},\boldsymbol{c})}[p_\beta(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})]$ respectively. Then the final objective is to minimize $L'_c(\phi, \theta) + \lambda L'_g(\phi, \beta)$.

3 Theoretical Justification

The goal of generative modeling is to learn the generated distribution $p_G(\boldsymbol{x})$ that is close to the real data distribution $p_r(\boldsymbol{x})$. Our proposed formulation in (2), however, does not explicitly optimize a distance measure between p_G and p_r, so it is unclear whether our method can match the distributions in the data space. In this section, we provide justification for the proposed formulation (2) by connecting it with the above goal, based on the analysis of WAE (Tolstikhin et al., 2018).

Let $\mathbb{P}_G$ and $\mathbb{P}_r$ be the induced probability measures of $p_G(\boldsymbol{x})$ and $p_r(\boldsymbol{x})$ respectively. We have the Kantorovich's formulation of the optimal transport (OT) problem with the L_1 cost:

$$W_1(\mathbb{P}_r, \mathbb{P}_G) = \inf_{\Gamma \in \mathcal{P}(\boldsymbol{x}\sim\mathbb{P}_r, \boldsymbol{y}\sim\mathbb{P}_G)} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\Gamma}[c(\boldsymbol{x}, \boldsymbol{y})],$$

[2] We have considered sampling multiple different neighborhoods within a given sequence as well, whose empirical performance was shown to be comparable with our proposed scheme with one local neighborhood, so we only reported the latter since it is simpler to implement.

TASK	UG		CG
DATASET	MSCOCO	WMTNEWS	ROCSTORY
VOCAB	27842	5728	20000
AVG LEN.	10.4	27.8	10.0
TRAIN S#	120K	278K	390K
DEV S#	-	-	50K
TEST S#	10K	10K	50K

Table 1: The statistics of the datasets. Avg Len. means the average length of sentences. S# refers to number of sentences. UG and CG stand for unconditional generation and conditional generation, respectively.

where $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_1$ is the cost function and $\mathcal{P}(\boldsymbol{x} \sim \mathbb{P}_r, \boldsymbol{y} \sim \mathbb{P}_G)$ is a set of all joint distributions of $(\boldsymbol{x}, \boldsymbol{y})$ with marginals $\mathbb{P}_r$ and $\mathbb{P}_G$ respectively. Note that $W_1(\mathbb{P}_r, \mathbb{P}_G)$ is also known as the 1-Wasserstein distance between $\mathbb{P}_r$ and $\mathbb{P}_G$. Then we have the following theorem which gives an upper bound of the 1-Wasserstein distance, whose proof is given in Appendix B.

Theorem 1. *Let $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ be a multivariate multinomial distribution with mean matrix $\bar{G}(\boldsymbol{z}) \in \mathbb{R}^{m \times n}$ which is a common choice for text modeling, i.e., each one-hot token $x_i|\boldsymbol{z}$ follows a multinomial with mean $\bar{G}_i(\boldsymbol{z}) \in$ simplex Δ^{m-1} for $i = 1, \ldots, n$. Then we have $W_1(\mathbb{P}_r, \mathbb{P}_G)$ is upper bounded by*

$$\inf_{q(\boldsymbol{z}|\boldsymbol{x}): q_z(\boldsymbol{z}) = p_\beta(\boldsymbol{z})} -2\mathbb{E}_{\boldsymbol{x} \sim p_r} \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})], \tag{4}$$

where $q_z(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x} \sim p_r(\boldsymbol{x})}[q(\boldsymbol{z}|\boldsymbol{x})]$ is the aggregated posterior and $p_\beta(\boldsymbol{z})$ is the implicit prior.

Hence by minimizing (4) with respect to θ and β, we learn the composite generator $G_\theta(g_\beta(\epsilon)) : \mathcal{E} \to \mathcal{X}$ that minimizes an upper bound of $W_1(\mathbb{P}_r, \mathbb{P}_G)$, which is consistent with the standard goal of generative modeling. However, this optimization problem is generally intractable due to the equality constraint and the nonparametric nature. Our formulation (2) can be regarded as an approximate problem of it by parametrizing $q(\boldsymbol{z}|\boldsymbol{x})$ with a distribution family induced by a stochastic encoder mapping E_ϕ, and relaxing the hard constraint $q_z(\boldsymbol{z}) = p_\beta(\boldsymbol{z})$ by introducing the relative entropy regularization $D_{\mathrm{KL}}(q_z(\boldsymbol{z}) \| p_\beta(\boldsymbol{z}))$.

4 Experiment Settings

4.1 Datasets

We conduct our experiments on three benchmark datasets, **MSCOCO** (Lin et al., 2014), **WMTNEWS** (Guo et al., 2018), and **ROCSTORY** (Mostafazadeh et al., 2016). All of the preprocessing steps are the same as Chen et al. (2018) and Wang and Wan (2019). The statistics of the resulting datasets are reported in Table 1.

4.2 Baselines

Unconditional Generation Three simplified variants of TILGAN are implemented for comparison:

- **TILGAN$_{\mathrm{P}}$**: a plain baseline using our backbone model, that is, a Transformer autoencoder and a GAN in the latent space based on KL divergence.
- **TILGAN$_{\mathrm{E}}$**: TILGAN$_{\mathrm{P}}$ equipped with decoder enhancement.
- **TILGAN$_{\mathrm{MD}}$**: TILGAN$_{\mathrm{P}}$ with the multi-scale discriminator.

In addition, the following existing models are adopted: recurrent neural network language model (RNNMLE), SeqGAN (Yu et al., 2017), RankGAN (Lin et al., 2017), GSGAN (Kusner and Hernández-Lobato, 2016), LeakGAN (Guo et al., 2018), textGAN (Zhang et al., 2017), FMGAN (Chen et al., 2018), ARAE (Zhao et al., 2018), Transformer language model (TMLE) (Vaswani et al., 2017).

Conditional Generation For conditional generation, we compare our model with Transformer (Vaswani et al., 2017), IE+MSA (Guan et al., 2019), Seq2Seq (Bahdanau et al., 2015), HLSTM (Li et al., 2015), CVAE (Sohn et al., 2015), and T-CVAE (Wang and Wan, 2019).

4.3 Automatic Evaluation Metrics

Unconditional Generation

- **TESTBLEU** (Yu et al., 2017): a quality metric comparing the n-gram similarity between generated samples and the whole test set.
- **SELFBLEU** (Zhu et al., 2018): a diversity metric calculating the similarity between one generated sentence and the whole remaining generation. The lower the SelfBLEU score is, the higher diversity we obtain in the generation.

Specifically, following Chen et al. (2018), we report BLEU-2/3/4/5 for TestBLEU and BLEU-2/3/4 for SelfBLEU.

Conditional Generation

- **BLEU** (Papineni et al., 2002): the BLEU score is calculated by taking the geometric mean of the n-gram BLEU scores where n is from 1 to 4.
- **DIVERSITY** (Li et al., 2016): the proportion of distinct n-grams in the generated results which evaluates the degree of diversity. D1 and D2 are reported for unigram and bigram, respectively.

Methods	(a) MSCOCO SelfBLEU B2%	B3%	B4%	TestBLEU B2%	B3%	B4%	B5%	Human Q	D	(b) WMTNews SelfBLEU B2%	B3%	B4%	TestBLEU B2%	B3%	B4%	B5%	Human Q	D
RNNMLE	75.4	51.1	23.2	82.0	60.7	38.9	24.8	3.33	2.8	66.4	33.7	**11.3**	76.1	46.8	23.1	11.6	3.65	2.8
SeqGAN	80.7	57.7	27.8	82.0	60.4	36.1	21.1	3.86	2.8	72.8	41.1	13.9	63.0	35.4	16.4	8.7	3.29	3.4
RankGAN	82.2	59.2	28.8	85.2	63.7	38.9	24.8	3.26	2.6	67.2	34.6	11.8	77.4	48.4	24.9	13.1	2.98	3.8
GSGAN	78.5	52.2	23.0	81.0	56.6	33.5	19.7	3.15	2.4	68.2	41.0	23.1	72.3	44.0	21.0	10.7	3.39	2.6
LeakGAN	91.2	82.5	68.9	92.2	79.7	60.2	41.6	3.07	3.0	85.7	69.6	37.3	92.0	72.5	50.2	32.1	2.51	2.8
textGAN	80.6	54.8	21.7	91.0	72.8	48.4	30.6	3.55	2.8	80.6	54.8	28.7	77.7	52.9	30.5	16.1	3.43	3.2
FMGAN	83.1	63.2	32.5	94.2	81.2	61.8	41.4	4.06	2.2	83.1	68.2	38.5	**93.2**	77.1	55.2	39.9	3.40	3.2
ARAE	63.2	41.6	19.1	86.7	69.3	44.2	24.5	2.93	3.0	53.4	30.4	17.3	84.4	62.9	39.8	22.0	2.29	2.6
TMLE	70.6	47.6	27.3	92.8	81.9	56.2	33.1	3.75	3.4	61.3	43.7	25.1	87.5	74.8	44.2	26.4	3.31	3.6
TILGAN	**61.6**	**35.6**	**9.9**	96.7	90.3	77.2	**53.2**	**4.38**	**3.8**	66.3	44.5	28.0	92.9	**81.7**	**61.7**	**40.7**	**3.81**	**4.0**
TILGAN$_{P}$	61.7	45.9	18.2	94.7	86.6	63.1	39.9	-	-	64.8	48.2	34.9	88.9	76.5	56.5	27.5	-	-
TILGAN$_{E}$	70.5	50.1	28.7	**98.8**	**94.5**	**81.3**	52.5	-	-	62.7	43.3	23.0	91.5	79.2	59.6	33.9	-	-
TILGAN$_{MD}$	63.5	38.7	11.8	95.1	84.9	64.8	44.1	-	-	**53.1**	**33.2**	20.6	92.6	78.2	53.9	29.5	-	-
IMP_POST	73.3	63.8	47.7	95.5	87.1	69.8	31.5	-	-	71.2	59.7	47.1	73.1	68.2	46.6	20.1	-	-
JSGAN	76.7	67.9	51.8	75.1	54.2	32.1	11.0	-	-	64.7	49.9	39.6	80.8	64.9	41.1	14.2	-	-
WGAN	90.4	80.9	69.0	73.0	53.8	34.2	12.5	-	-	93.3	91.0	88.6	89.1	77.4	50.2	26.7	-	-

Table 2: SelfBLEU and TestBLEU results on MSCOCO and EMNLP WMTNews datasets. Q and D denote the quality and diversity evaluated by human, respectively. The results of previous baselines are listed in the top region, our method TILGAN together with simplified variants are shown in the middle, and more variants for further ablation studies are at the bottom. The bold numbers are the best results in each column.

- **Adversarial Success** (Li et al., 2017): the fraction of instances in which a model is capable of fooling a fine-tuned BERT classifier with the above 95% accuracy on the development set of the classification task. Higher values are better. The positive examples are original stories and negative examples are stories consisting of a random sentence from another story.

4.4 Implementation

Unconditional Generation We implement a Transformer-based autoencoder with 2 layers, 4 heads, 512 embedding dimensions, and 512 hidden dimensions. The generator and discriminator are implemented by 3 layers multi-layer perceptron (MLP). We set the maximum sequence length to 15 and 32 for MSCOCO and WMTNews, respectively. During training, each sentence is padded to the maximum length when fed into the encoder, and then the encoder produces a latent vector for every input token. During testing, the latent-generator generates a sequence of latent vectors with the same maximum length, and then, conditional on the latent vectors, the decoder generates a sentence which ends when a special token, <EOS>, is generated. We adopt Adam (Kingma and Ba, 2015) as the optimizer with a learning rate of 0.00025 and 0.0001 for autoencoder and GAN structure, respectively with a dropout rate of 0.3.

Conditional Generation We adopt the same Transformer encoder-decoder architecture as the backbone model and similar setups as Wang and Wan (2019). The Transformer structure has 6 layers, 8 self-attention heads, 512 dimensions for hidden states, and uses shared attention layers for encoder and decoder which allows the decoder to attend to the encoder state and the decoder state at the same time to make the completed story more coherent. The generator has 3 layers and the discriminator has 4 layers. We adopt Adam (Kingma and Ba, 2015) as the optimizer with a learning rate of 0.0001 and a dropout rate of 0.15.

More details of the experimental setup and hyperparameter settings are shown in the Appendix A.

5 Experimental Results

5.1 Unconditional Generation

• **Generation Quality** The first experiment is to compare the quality of generated sentences of different models. In general, as shown in Table 2, TILGAN outperforms all baseline models in TestBLEU on both MSCOCO and WMTNews datasets, which clearly indicates the advantages of our proposed framework. We make five main observations. Firstly, we notice that TMLE is comparable to most GAN baselines, which shows the powerful fitting capacity of the Transformer architecture as well as the inferior performance of previous GAN implementations. Despite this, TILGAN$_{P}$ outperforms TMLE by a wide margin, demonstrating that our backbone model combining a Transformer autoen-

coder and a GAN can not only take advantage of the Transformer's capacity, but also exhibit benefits from our GAN formulation to further boost the performance. Furthermore, compared with TILGAN$_{\text{P}}$, our full version TILGAN achieves an improvement of 13.3% and 13.2% for TestBLEU5 on MSCOCO and News, respectively. This confirms the effectiveness of the proposed multi-scale discriminators and decoder enhancement. In detail, when comparing the simplified variants TILGAN$_{\text{E}}$ v.s. TILGAN$_{\text{MD}}$, the improvement of TILGAN$_{\text{E}}$ over the plain baseline TILGAN$_{\text{P}}$ is larger than that of TILGAN$_{\text{MD}}$, which illustrates that incorporating TILGAN$_{\text{E}}$ is more crucial to improving the generation quality. Lastly, we compare TILGAN with all previous methods and observe an average improvement of 6.3% for TestBLEU5 on two datasets against the previous state-of-the-art FMGAN, which suggests the superiority of our method.

• **Generation Diversity** The generation diversity is evaluated by SelfBLEU scores, which are shown in Table 2. First, compared with the baselines with comparable and worse TestBLEU, e.g., FMGAN and LeakGAN, our TILGAN achieves lower SelfBLEU scores, which indicates a better quality-diversity trade-off from TILGAN. We notice that RNNMLE achieves the best SelfBLEU score on WMTNews but its quality shown by TestBLEU is pretty low and tends to generate incoherent or meaningless segments, which can be confirmed by the generated samples in Appendix C. In addition, from the results of TILGAN$_{\text{MD}}$, we find that incorporating the multi-scale discriminator leads to a significant drop in SelfBLEU, suggesting that most of the performance gains in generation diversity are attributed to our design of multi-scale discriminators in contrast to decoder enhancement. Moreover, when comparing the performance across two datasets, we find that the SelfBLEU scores of our models are lower on MSCOCO than that on WMTNews, illustrating that it is easier to generate more diverse texts on MSCOCO. One possible reason is that the texts in MSCOCO are shorter than the texts in WMTNews as shown in Table 1. When generating long sequences, models are prone to generate repeated tokens and phrases. The same phenomenon was also observed for many other baseline models like ARAE and FMGAN.

5.2 Conditional Generation

In addition to unconditional generation, we test our model in a story completion task to verify its ability in conditional generation. Table 3 shows the automatic metrics in four metrics, with several observations. (i) Overall, among all models, TILGAN achieves new state-of-the-art results on the ROCStory dataset, showing the superiority of our method. (ii) Our model obtains substantial improvement in the quality metrics of generated answers, with 0.32% gains in BLEU, 6.92% gains in the adversarial success. It demonstrates that the generated plots are in high coherence, which not only share a higher proportion of word overlap with ground-truth answers, but also have a higher success rate fooling the BERT classifier. (iii) As for diversity, TILGAN improves upon the state-of-the-art methods from 3.63% to 3.88% on D1 and 23.46% to 25.61% on D2, showing that TILGAN produces stories consisting of more diverse and distinct n-grams.

5.3 Human Evaluation

Due to the limitations of automatic evaluation metrics, we invite 5 judges to rate 100 sentences generated by different models on a scale from 1 to 5 for both unconditional and conditional generation tasks. The results for unconditional generation are shown in Table 2. TILGAN shows a superior performance, which confirms that our model is able to generate more realistic samples than the baseline models on two datasets. Among all baseline models, FMGAN has a high quality score but a low diversity score, which indicates that most of its generated samples are repeated sentences that lack diversity. Additional evidence is shown through the case study in Section 6.2.

In addition, the human evaluation results on story completion are shown in Table 3 where we only compare TILGAN with the best baseline, i.e., T-CVAE. We use Gram metric to evaluate whether the generated story plot proceeds naturally, and Logic metric to evaluate whether the plot is reasonable and coherent following Wang and Wan (2019). Compared with T-CVAE, TILGAN is better in both Gram and Logic, demonstrating that the generated story plots of TILGAN are more natural and coherent.

METHODS	BLEU%	D1%	D2%	AS%	GRAM	LOGIC
SEQ2SEQ	2.90	2.69	15.95	80.97	-	-
HLSTM	2.31	2.63	14.80	72.46	-	-
CVAE	3.03	2.72	16.32	81.18	-	-
TRANS.	3.05	2.93	16.75	82.51	-	-
T-CVAE	4.25	3.63	23.46	87.54	3.32	3.24
TILGAN	**4.57**	**3.88**	**25.61**	**94.46**	3.58	3.60

Table 3: Results on story completion task. AS refers to adversarial success score. TRANS. denotes a vanilla Transformer model.

6 Analyses

6.1 Ablation Study

To examine the impact of KL loss and the implicit prior, we conduct ablation studies of different designs. We construct three variants of TILGAN and conduct experiments on two datasets of unconditional generation, whose results are shown in the bottom region of Table 2.

• **Impact of KL Loss** First, we implement two variants named JSGAN (Goodfellow et al., 2014) and WGAN (Arjovsky et al., 2017) by replacing the KL loss term with JS divergence and Wasserstein distance, while keeping the same architectures. In general, TILGAN_P outperforms JSGAN and WGAN in terms of SelfBLEU and TestBLEU on two datasets. Particularly, it is observed that with KL loss, the SelfBLEU4 score drops from 51.8% and 69.0% to 18.2% over JSGAN and WGAN on MSCOCO, and similar downward trends are observed on WMTNews. It demonstrates that minimizing KL loss indeed benefits the generation diversity, which is consistent with previous findings in Shen et al. (2020). In addition, the TestBLEU of TILGAN_P achieves an improvement of 28.9% and 27.4% for TestBLEU5 on MSCOCO over JSGAN and WGAN, respectively.

• **Implicit Prior v.s. Implicit Posterior** In addition to imposing an implicit prior, one can instead impose an implicit posterior as well by moving the transformation network of the latent-generator to the encoder and leaving a Gaussian prior. This results in a variant with nearly the same total number of parameters, named IMP_POST. We see from Table 2 that IMP_POST performs worse than TILGAN_P with an implicit prior, suggesting that enlarging the distribution family of posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ contributes less to improving the overall generation performance than enlarging that of prior $p_z(\boldsymbol{z})$, which is consistent to the analysis in the second paragraph of Section 2.1.

6.2 Case Study

To further analyze the real quality and diversity of the generated sentences, some are examined and presented in Table 4 and more examples are shown in Appendix C. First, the samples generated by TILGAN are more coherent and semantically meaningful. The majority of texts of TILGAN are in subject–verb–object order while those of other models are not. In addition, TILGAN exhibits more diverse sentence structures and word choices than others. For example, although each sentence generated by FMGAN looks good in quality, there are many repeated sentences or phrases, leading to a low diversity. The case study is consistent with the human evaluation results in Section 5.3.

7 Related Work

Conventional text generation models leverage maximum likelihood estimation (MLE) with teacher forcing and have shown powerful generation capabilities (Mikolov et al., 2010; Cho et al., 2014; Bahdanau et al., 2016; Radford et al., 2019; Brown et al., 2020) but they suffer from the exposure bias problem. To address this, several solutions were introduced including scheduled sampling (Bengio et al., 2015), professor forcing (Lamb et al., 2016), and Gibbs sampling (Su et al., 2018).

GAN-based text generation methods can be categorized into three classes: reinforcement learning (RL) based methods, Gumbel-Softmax (GS) based methods and latent feature matching methods. RL-based methods (Yu et al., 2017; Lin et al., 2017; Che et al., 2017; Guo et al., 2018; Fedus et al., 2018) design a reward incorporated with the discriminators, and use policy gradient or actor-critic approaches to update the generator to resolve the gradient propagating issue over discrete tokens. However, they suffer from high variance and mode collapse issues caused by the unstable policy gradient training process and the lack of a reliable guiding signal (Zhang et al., 2017; Chen et al., 2018). GS-based methods (Kusner and Hernández-Lobato, 2016) apply Gumbel-Softmax which is a continuous relaxation technique for transforming the output of a generator to be as close to one-hot as possible in order to make the samples from a discrete distribution like a multinomial differentiable with respect to the distribution parameters.

Latent feature matching methods (Zhang et al., 2017; Zhao et al., 2018) learn a manifold in the latent space instead of the discrete output space.

RankGAN:	(1) *A blue blue train sits on tracks with his residential asian toys.* (2) *A reflection of two birds walking by a sidewalk.*
FMGAN:	(1) *A man is standing on a table with a dog.* (2) *A man is standing on a table with a dog on a field.* (3) *A man is standing on a field of a large building.*
TILGAN:	(1) *A little boy sitting on a bench with a little girl.* (2) *A blue and white public transit bus is driving down acity street.* (3) *A train is going down the tracks in a forest.*

Table 4: Examples of generated sentences from RankGAN, FMGAN and our model.

This kind of methods usually incorporates an autoencoder to build the feature space and force the generator's latent output distribution to approach the real data latent distribution. Our method also resides in this category. To ease adversarial training, Zhang et al. (2017) introduce adversarial feature matching method by incorporating a kernelized discrepancy metric to match high-dimensional latent representations of real and synthetic sentences. ARAE (Zhao et al., 2018) extends AAE (Makhzani et al., 2015) to model discrete sequences and learns a parameterized prior by a generative model trained with WGAN. In contrast to our TILGAN whose Transformer-based encoder and decoder are both stochastic, ARAE uses RNN-based encoder and decoder which are both deterministic, as required in their theory, which reduces the model expressiveness and results in much poorer performance than ours as shown in Table 2. iVAE (Fang et al., 2019) proposes a VAE (Kingma and Welling, 2014) with an implicit posterior which is inferior to the implicit prior that we adopt according to the ablation study in Section 6.1. WAE-S (Bahuleyan et al., 2019) is a WAE(Tolstikhin et al., 2018) with a stochastic encoder trained using MMD with a distinct goal of improving the reconstruction ability.

8 Conclusion

In this paper, we proposed Transformer-based Implicit Latent GAN (TILGAN), for text generation. It combines a Transformer autoencoder and a GAN through matching the distributions of multi-token sequences in the Transformer's latent space based on KL divergence. To improve the local and global coherence, we introduced a multi-scale discriminator to utilize the semantic information on varying scales. To train the decoder reliably, we enhanced the objective function by another KL term, forcing the decoder to be compatible with the latent-generator. We theoretically connected the proposed formulation with the standard goal of generative modeling. Empirically, TILGAN achieved the state-of-the-art performance on three widely used datasets for unconditional tasks and story completion task, which demonstrated the effectiveness of our method to generate texts of high quality and diversity compared with the existing approaches.

Acknowledgments

This work was supported by the General Research Fund (GRF) of Hong Kong (No. 16201320). Y. Song was supported by NSFC under the project "The Essential Algorithms and Technologies for Standardized Analytics of Clinical Texts" (12026610) and Shenzhen Institute of Artificial Intelligence and Robotics for Society under the project "Automatic Knowledge Enhanced Natural Language Understanding and Its Applications" (AC01202101001). The authors also want to thank the anonymous reviewers for their valuable comments and suggestions.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 214–223.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end Attention-based Large Vocabulary Speech Recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE.

Hareesh Bahuleyan, Lili Mou, Hao Zhou, and Olga Vechtomova. 2019. Stochastic Wasserstein Autoencoder for Probabilistic Sentence Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4068–4076.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, pages 1171–1179.

Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. 2020. High Fidelity Speech Synthesis with Adversarial Networks. In *International Conference on Learning Representations*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*.

Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-Likelihood Augmented Discrete Generative Adversarial Networks. *arXiv preprint arXiv:1702.07983*.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial Text Generation via Feature-Mover's Distance. In *Advances in Neural Information Processing Systems*, pages 4666–4677.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit Deep Latent Variable Models for Text Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3937–3947.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. MaskGAN: Better Text Generation via Filling in the _. In *International Conference on Learning Representations*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, pages 2672–2680.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story Ending Generation with Incremental Encoding and Commonsense Knowledge. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019*.

J Guo, S Lu, H Cai, W Zhang, Y Yu, and J Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Matt J Kusner and José Miguel Hernández-Lobato. 2016. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *arXiv preprint arXiv:1611.04051*.

Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor Forcing: A New Algorithm for Training Recurrent Networks. In *Advances in Neural Information Processing Systems*, pages 4601–4609.

Hung Le, Doyen Sahoo, Nancy Chen, and Steven Hoi. 2019. Multimodal Transformer Networks for End-to-End Video-Grounded Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5612–5623.

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 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference*

on Natural Language Processing (Volume 1: Long Papers), pages 1106–1115.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial Ranking for Language Generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European conference on computer vision*, pages 740–755. Springer.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. In *Eleventh annual conference of the international speech communication association*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Xinwei Shen, Tong Zhang, and Kani Chen. 2020. Bidirectional Generative Modeling Using Adversarial Gradient Estimation. *arXiv preprint arXiv:2002.09161*.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 28, pages 3483–3491. Curran Associates, Inc.

Jinyue Su, Jiacheng Xu, Xipeng Qiu, and Xuanjing Huang. 2018. Incorporating Discriminator in Sentence Generation: a Gibbs Sampling Method. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2018. Wasserstein auto-encoders. In *International Conference on Learning Representations*.

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 Neural Information Processing Systems*, pages 5998–6008.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning Deep Transformer Models for Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822.

Tianming Wang and Xiaojun Wan. 2019. T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5233–5239.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural Text Generation With Unlikelihood Training. In *International Conference on Learning Representations*.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306.

Qingyang Wu, Lei Li, and Zhou Yu. 2020. TEXTGAIL: Generative Adversarial Imitation Learning for Text Generation. *arXiv preprint arXiv:2004.13796*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial Feature Matching for Text Generation. In *Proceedings of the 34th International Conference on Machine Learning*, pages 4006–4015.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially Regularized Autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5902–5911.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

A Reproducibility Checklist

- **Description of Computing Infrastructure** Tesla V100S-PCIE-32GB
- **Average Runtime**

MODEL	UG		CG
DATASET	MSCOCO	WMTNEWS	ROCSTORY
TARAE	5.8	9.2	-
ENHANCED	7.5	11	-
LOCALD	8	12	-
TILGAN	8.5	13.4	106

Table 5: The average runtime per epoch for each model, estimated in minutes. UG and CG refer to unconditional generation and conditional generation, respectively.

- **Number of Parameters**

MODEL	UG	CG
TILGAN	25.4M	30.8M

Table 6: The number of parameters of each model. UG and CG refer to unconditional generation and conditional generation, respectively.

- **Validation Performance** No validation evaluation for unconditional generation and conditional generation tasks.
- **Number of Runs** We conduct 60 runs for unconditional generation tasks and 30 runs for conditional generation tasks.
- **Bounds and Best Setting for Hyperparameters** Please refer to Table 3 for unconditional generation task and Table 4 for conditional generation task.

	MSCOCO		WMTNews	
	Bound	Best-performing	Bound	Best-performing
max len	15	15	32	32
batch size	[32,256]	256	[32,256]	256
emb size	[256,1024]	512	[256,1024]	512
hidden size	[256,1024]	512	[256,1024]	512
num layers	[1,6]	2	[1,6]	2
num heads	4	4	4	4
squeezed hidden size	[28,256]	56	[28,256]	56
noise size	[50,512]	100	[50,512]	100
niters autoencoder	[1,3]	1	[1,3]	1
niters discriminator	[1,3]	1	[1,3]	1
niters enhanceD	[1,3]	1	[1,3]	1
niters generator	[1,3]	1	[1,3]	1
niters gan into encoder	[1,3]	1	[1,3]	1
learning rate autoencoder	[0.01,10]	0.08	[0.01,10]	0.24
learning rate gan encoder	[1e-5,1e-2]	1e-4	[1e-5,1e-2]	1e-4
learning rate generator	[1e-5,1e-2]	1e-4	[1e-5,1e-2]	1e-4
learning rate discriminator	[1e-5,1e-2]	1e-4	[1e-5,1e-2]	1e-4

Table 7: The bounds for each hyperparameter and best-performing setting for unconditional generation task.

	ROCStory	
	Bound	Best-performing
num layers	[1,8]	6
hidden size	[256,1024]	512
num heads	[4,12]	8
emb size	300	300
latent dimension	[32,256]	64
batch size	[32,128]	64
learning rate	[1e-5,1e-2]	1e-4
droupout rate	[0,0.5]	0.15

Table 8: The bounds for each hyperparameter and best-performing setting for conditional generation task.

B Proof of Theorem 1

Proof of Theorem 1. In this proof, we let $\boldsymbol{x}$ be the real data, $\boldsymbol{y}$ be the generated data, and $\boldsymbol{z}$ be the latent variable. Let $p_G(\boldsymbol{y}, \boldsymbol{z}) = p_\beta(\boldsymbol{z})p_\theta(\boldsymbol{y}|\boldsymbol{z})$ be the joint distribution of $(\boldsymbol{y}, \boldsymbol{z})$, where $\boldsymbol{z}$ is sampled from prior $p_\beta(\boldsymbol{z})$ and then $\boldsymbol{y}$ is sampled from the decoder conditional $p_\theta(\boldsymbol{y}|\boldsymbol{z})$. Further let $\mathcal{P}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}$ denote the set of all joint distributions of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ such that $\boldsymbol{x} \sim p_r(\boldsymbol{x})$, $(\boldsymbol{y}, \boldsymbol{z}) \sim p_G(\boldsymbol{y}, \boldsymbol{z})$, and $\boldsymbol{x} \perp\!\!\!\perp \boldsymbol{y}|\boldsymbol{z}$; let $\mathcal{P}_{\boldsymbol{x},\boldsymbol{z}}$ be the set of marginal distributions of $(\boldsymbol{x}, \boldsymbol{z})$ induced by $\mathcal{P}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}$, that is, the set of distributions with marginals $\boldsymbol{x} \sim p_r(\boldsymbol{x})$ and $\boldsymbol{z} \sim p_\beta(\boldsymbol{z})$.

Recall that n is the sequence length and m is the number of words in the vocabulary. For the i-th word x_i which is an m-dimensional one-hot vector, indicator $\mathbf{1}(x_i = j) = 1$ if the j-th dimension of x_i is equal to 1 and $\mathbf{1}(x_i = j) = 0$ otherwise, for $j = 1, \ldots, m$. Then we have

$$
\begin{aligned}
W_1(\mathbb{P}_r, \mathbb{P}_G) \leq W_1^\dagger(\mathbb{P}_r, \mathbb{P}_G) &:= \inf_{p \in \mathcal{P}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{z}\sim p_\beta(\boldsymbol{z})}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}\mathbb{E}_{\boldsymbol{y}\sim p_\theta(\boldsymbol{y}|\boldsymbol{z})}[c(\boldsymbol{x}, \boldsymbol{y})] \\
&= \inf_{p \in \mathcal{P}_{\boldsymbol{x},\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{z}\sim p_\beta(\boldsymbol{z})}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}\left[2\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{1}(x_i = j)(1 - \bar{G}_{ij}(\boldsymbol{z}))\right] \\
&< \inf_{p \in \mathcal{P}_{\boldsymbol{x},\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{z}\sim p_\beta(\boldsymbol{z})}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}\left[-2\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{1}(x_i = j)\ln \bar{G}_{ij}(\boldsymbol{z})\right] \\
&= \inf_{q(\boldsymbol{z}|\boldsymbol{x}):q_z(\boldsymbol{z})=p_\beta(\boldsymbol{z})} \mathbb{E}_{\boldsymbol{x}\sim p_r}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[-2\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{1}(x_i = j)\ln \bar{G}_{ij}(\boldsymbol{z})\right] \\
&= \inf_{q(\boldsymbol{z}|\boldsymbol{x}):q_z(\boldsymbol{z})=p_\beta(\boldsymbol{z})} \left\{-2\mathbb{E}_{\boldsymbol{x}\sim p_r}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})]\right\},
\end{aligned}
$$

where the first inequality comes from Tolstikhin et al. (2018, eq. 9), and the second inequality is due to the fact that $1 - l < -\ln l$ for all $l \in (0, 1)$, leading to the desired result. □

C Generated Examples

Table 9: Generated samples on ROCStory dataset.

story:	_____. when i got to the stop sign , the check engine light started flashing . i panicked and carefully drove the van to the nearest mechanic shop . they checked it out but could not repair the van . the van had to be sold for parts and i had to get a new vehicle .
T-CVAE:	i was driving my van down the street one day .
TILGAN:	i was driving my van to work one day .
story:	krista was organizing her office ._____ . they were big and heavy . she assembled them carefully . when she put all her books on them , they collapsed !
T-CVAE:	she had a bunch of books .
TILGAN:	she bought some new books .
story:	the man won a contest . he went to the station to collect . _____ . he did n't really like the band . he tried to sell them back to the radio employees .
T-CVAE:	he got a ticket for a band .
TILGAN:	he saw some band members .
story:	billy is bored . billy sits with his friends thinking of something to do . billy suggest they all head to the lake to go fishing . _____ . billy takes his friends to go fishing and has great time .
T-CVAE:	billy and his friends go fishing together .
TILGAN:	billy and his friends go fishing .
story:	_____ . her house was full of dust . she could n't believe how filthy it was . alicia then decided to clean it . when she was done cleaning and it sparkled .
T-CVAE:	alicia was in the basement .
TILGAN:	alicia was cleaning her house .

Table 10: Generated samples on MSCOCO dataset.

TextGAN: - a train traveling down a street . - a train station . - a street sign on a street . - is in a bathroom with a sink controls .	**SeqGAN**: - a red stop sign . - a couple of people are walking on a log and trees . - the train car traveling mannequin driving down the tracks . - people standing next to a large building .
RankGAN: - a blue blue train sits on tracks with his residential asian toys . - a reflection of two birds walking by a sidewalk . - a man tourist train in the egret - a white fire hydrant stands next to each other .	**MLE**: - an orange booth contains the in traffic light under a sign - a man with hat on the horse in the street - a couple walking around city tracks with people - the bird are walking next to a small blue coop
LeakGAN: - a table topped with pots . . - a bathroom with a glass shower , sink , toilet and sink . - a woman wearing a glass is sitting on a cupboard . - a group of men talking .	**FM-GAN**: - a man is standing on a skateboard on the beach - a man on a tennis game with a kite - a man on a table with a red and white and a building - a man is standing on a table with a dog
ARAE: - a city street sign in the park bench parked in the group group the man - two people standing at motorcycles on the bench - two people standing at motorcycles on the bench at white kitchen - there is a city bus on their city street sign parked in blue blue bus - a white plane on the air plane parked in snow group their plane parked	**TILGAN**: - a little boy sitting on a bench with a little girl - a group of people in the middle of a field - a large passenger plane flying through the sky - a woman is sitting in a kitchen next to a restaurant - a small bird sitting on a branch of a tree

Table 11: Generated samples on EMNLP WMT dataset.

SeqGAN: - it said the yield on our most traveled to china ' s capital for " the annual bank of cost credit against cuba . - the cars (, taiwan argues that the cease - fire contained included already reported on the outlook . - " both the republican leader in the years of leadership she was in rome and signed a close cabinet , " she said . - russia has said in its twitter documents since january which demanded that lawmakers had clearly been involved about .
RankGAN: - the lakers left the us to hope of the office and chose the general administration to build a further mission . - ms . bush had been remembered in a red ring after it taking these sunday week made focusing himself against the number of games . - " " i ' d thought something i am running everything i saw my own american life usually respect at all , as some equipment they have that . - and i was hoping that management does still even bring to hillary carson , and listen to a guy playing off back .
MLE: - what we need do this case if anybody had now touched a in - town community , all your kids in exchange barriers are needed in , no more , all may come out the coming in reflect the options . - so the scottish government is significant until not extension you own very so hard to change my job for your six points at half , social opinion and take beyond . - us president - elect donald trump will re consider an effort to set out that it would be to accept from the us - city solution to the world . - local judges ask for her children and went into a video itself , she said for 2016 ' s early next day , he said .
LeakGAN: - picture west eight my might confidence , zero confidence my either nazi a a time having accounts , skills a difference x having must difference time having a develop pakistan confidence time time killed wilson partners nazi unfair zero phones develop vital confidence a might showed a having confidence develop a - pupils evidence accounts having confidence confidence theft abortion time time sized time west coming a unfair time affecting time my theft a a killed killed phones , , time questioned pakistan a partners evidence sized confidence unfair my eight time pakistan zero zero confidence partners either seventh having , killed a
GsGAN: - i hope that i do something like that it ' s a very important thing , i know what you want to see this i didn ' t know , i think it does not be able to work out this way that ' s not a lot better needs to - the actor is it , well , which was a good job in a writing - christmas time out of a three - year - old woman who had been charged for the murder , he said that he was investigating the government ' s decision to 19 . 6 million - to give the first time , it added , he had a few days and now he ' s not just a new administration , he will do the same time before . - it is that , but the two - year - old woman he didn ' t agree on : they had been a right ago because i wanted to have to do something i was trying to kill them , i am , but i can do it had to stay
FM-GAN: - The United States , the United States has been a major group of the United States in the United States , the United States in the United States . - We have to be able to pay the money to be able to pay the money to be able to pay the money to pay for the same time ," he said . - " It ' s a lot of people who have been a woman , and I have been told the police ," she said . - The man ' s death was a " bit of the incident , but the police said that the police had been taken to the city , and the police officer was a " very dangerous - driving area . - We have to be able to do the government to be able to do the government to be able to leave the country ," he said .
ARAE: - a more . 5 per cent the company said it would not be expected to rise if he hit the 2 percent year , it said , rose 2 , 500 ,when the only reason only be the best way for the best time for the best time for the best time for them and not being able to have done with much - the fact : the fact only now not being able to have done with a much more time for the age amount time with a much time for the best time for - the fact the only reason only be the best way . - the fact : the fact only now being a more person with a person with each person with a much time with the best time for them as much as a person
TILGAN: - many people who died , although they didn ' t have been on the same day , not just because of those who had been out of them . - " i had to be able to get a good deal with the right time ," he said in a statement . - that ' s why , in my life is now that ' s not the same thing , and how much money is . - we are still working closely with the community who is still in the world , but we can ' t be the best . - we can ' t get some good players in the league , but not only because we ' ve played well .