Improving Adversarial Text Generation by Modeling the Distant Future

Ruiyi Zhang[1], **Changyou Chen**[2], **Zhe Gan**[3], **Wenlin Wang**[4]
Dinghan Shen[3], **Guoyin Wang**[1], **Zheng Wen**[5], **Lawrence Carin**[1]
[1] Duke University [2] University at Buffalo [3] Microsoft Dynamics 365 AI
[4] Citadel LLC [5] DeepMind
ryzhang@cs.duke.edu

Abstract

Auto-regressive text generation models usually focus on local fluency, and may cause inconsistent semantic meaning in long text generation. Further, automatically generating words with similar semantics is challenging, and hand-crafted linguistic rules are difficult to apply. We consider a *text planning* scheme and present a model-based imitation-learning approach to alleviate the aforementioned issues. Specifically, we propose a novel guider network to focus on the generative process over a longer horizon, which can assist next-word prediction and provide intermediate rewards for generator optimization. Extensive experiments demonstrate that the proposed method leads to improved performance.

1 Introduction

Text generation is an important area of investigation within machine learning. Recent work has shown excellent performance on a number of tasks, by combining reinforcement learning (RL) and generative models. Example applications include image captioning (Ren et al., 2017; Rennie et al., 2016), text summarization (Li et al., 2018b; Paulus et al., 2017; Rush et al., 2015), and adversarial text generation (Guo et al., 2017; Lin et al., 2017; Yu et al., 2017; Zhang et al., 2017; Zhu et al., 2018). The sequence-to-sequence framework (Seq2Seq) (Sutskever et al., 2014) is a popular technique for text generation. However, models from such a setup are typically trained to predict the next token given previous ground-truth tokens as input, causing what is termed exposure bias (Ranzato et al., 2016). By contrast, sequence-level training with RL provides an effective means of solving this challenge, by treating text generation as a sequential decision-making problem. By directly optimizing an evaluation score (cumulative rewards) (Ranzato et al., 2016), state-of-the-art results have been obtained in many text-generation tasks (Paulus et al., 2017; Rennie et al., 2016). However, one problem in such a framework is that rewards in RL training are particularly sparse, since a scalar reward is typically only available after an entire sequence has been generated. Furthermore, the recurrent models focus more on local fluency, and may cause inconsistent semantic meanings for long text generation.

For RL-based text generation, most existing works rely on a model-free framework, which has been criticized for its high variance and poor sample efficiency (Sutton and Barto, 1998). On the other hand, while model-based RL methods do not suffer from these issues, they are usually difficult to train in complex environments. Further, a learned policy is usually restricted by the capacity of an environment model. Recent developments on model-based RL (Gu et al., 2016; Kurutach et al., 2018; Nagabandi et al., 2017) combine the advantages of these two approaches, and have achieved improved performance by learning a model-free policy, assisted by an environment model. In addition, model-based RL has been employed recently to solve problems with extremely sparse rewards, with curiosity-driven methods (Pathak et al., 2017).

In this paper, we propose a *model-based imitation-learning* method to overcome the aforementioned issues in text-generation tasks. Our main idea is to employ an explicit guider network to model the generation environment in the feature space of sentence tokens, used to emit intermediate rewards by matching the predicted features from the guider network and features from generated sentences. The guider network is trained to encode global structural information of training sentences, and thus is useful to guide next-token prediction in the generative process. Within the proposed framework, to assist the guider network, we also develop a new type of self-attention mechanism to provide high-level planning-ahead information

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2516–2531
July 5 - 10, 2020.

and maintain consistent semantic meaning. Our experimental results demonstrate the effectiveness of proposed methods.

2 Background

Text Generation Model Text generation models learn to generate a sentence $Y = (y_1, \ldots, y_T)$ of length T, possibly conditioned on some context X. Here each y_t is a token from vocabulary $\mathcal{A}$. Starting from the initial state $\boldsymbol{s}_0$, a recurrent neural network (RNN) produces a sequence of states $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T)$ given an input sentence-feature representation $(e(y_1), \ldots, e(y_T))$, where $e(\cdot)$ denotes a word embedding function mapping a token to its d-dimensional feature representation. The states are recursively updated with a function known as the *cell*: $\boldsymbol{s}_t = h(\boldsymbol{s}_{t-1}, e(y_t))$. One typically assigns the following probability to an observation y at location t: $p(y|Y_{<t}) = [\text{softmax}(g(\boldsymbol{s}_t))]_y$. Together (g, h) specifies a probabilistic model π, *i.e.*,

$$\log \pi(Y) = \sum_t \log p(y_t|Y_{<t}). \tag{1}$$

To train the model π, one typically uses maximum likelihood estimation (MLE), via minimizing the cross-entropy loss, *i.e.*, $J_{\text{MLE}}(\pi) = -\mathbb{E}[\log \pi(Y)]$. In order to generate sentence Y^s from a (trained) model, one iteratively applies the following operations:

$$y^s_{t+1} \sim \text{Multi}(1, \text{softmax}(g(\boldsymbol{s}_t))), \tag{2}$$

$$\boldsymbol{s}_t = h(\boldsymbol{s}_{t-1}, e(y^s_t)), \tag{3}$$

where $\text{Multi}(1, \cdot)$ denotes one draw from a multinomial distribution.

Model-Based Imitation Learning Text generation can be considered as an RL problem with a large number of discrete actions, *deterministic* transitions, and *deterministic* terminal rewards. It can be formulated as a Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the deterministic environment dynamics, $r(\boldsymbol{s}, y)$ is a reward function, and $\gamma \in (0, 1)$ is the discrete-time discount factor. The policy π_ϕ, parameterized by ϕ, maps each state $\boldsymbol{s} \in \mathcal{S}$ to a probability distribution over $\mathcal{A}$. The objective is to maximize the expected reward:

$$J(\pi) = \sum_{t=1}^{\infty} \mathbb{E}_{P,\pi} \left[\gamma^{t-1} \cdot r(\boldsymbol{s}_t, y_t)\right]. \tag{4}$$

In model-based imitation learning (Baram et al., 2017; Cheng et al., 2019), a model is built to make predictions for future state $\boldsymbol{s}_{t+\triangle t}$ conditioned on the current state[1], which can be used for action selection, *e.g.*, next-token generation. This model is typically a discrete-time system, taking the current state-action pair $(\boldsymbol{s}_t, y_t)$ as input, and outputting an estimate of the future state $\boldsymbol{s}_{t+\triangle t}$ at time $t + \triangle t$. At each step t, y_t is chosen based on the model, and the model will re-plan with the updated information from the dynamics. This control scheme is different from a standard model-based method, and is referred to as *model-predictive control* (MPC) (Nagabandi et al., 2017). Note that in our setting, the state in RL typically corresponds to the current generated sentences $Y_{1,\ldots,t}$ instead of the RNN state of generator (decoder).

3 Proposed Model

The model is illustrated in Figure 1, with an autoeocoder (AE) structure for sentence feature extraction and generation. The encoder is shared for sentences from both training data and generated data, as explained in detail below. Overall, text generation can be formulated as an imitation-learning problem. At each timestep t, the agent, also called a generator (which corresponds to the LSTM decoder), takes the current LSTM state as input, denoted as $\boldsymbol{s}_t$. The policy $\pi_\phi(\cdot|\boldsymbol{s}_t)$ parameterized by ϕ is a conditional generator, to generate the next token (action) given $\boldsymbol{s}_t$, the *observation* representing the current generated sentence. The objective of text generation is to maximize the total reward as in (4). We detail the components for our proposed model in the following subsections.

3.1 The Guider Network

The guider network, implemented as an RNN with LSTM units, is adopted to model *environment dynamics* to assist text generation. The idea is to train a guider network such that its predicted sentence features at each time step are used to assist next-word generation and construct intermediate rewards, which in turn are used to optimize the sentence generator. Denote the guider network as $G^\psi(\boldsymbol{s}^G_{t-1}, \boldsymbol{f}_t)$, with parameters ψ and input arguments $(\boldsymbol{s}^G_{t-1}, \boldsymbol{f}_t)$ at time t, to explicitly write out the dependency on the *guider network* latent state $\boldsymbol{s}^G_{t-1}$ from the previous time step. Here $\boldsymbol{f}_t$ is the input to the LSTM guider, which represents the feature of the current generated sentence extracted

[1] $\triangle t > 1$; the model predicts future states based on the collected trajectories.

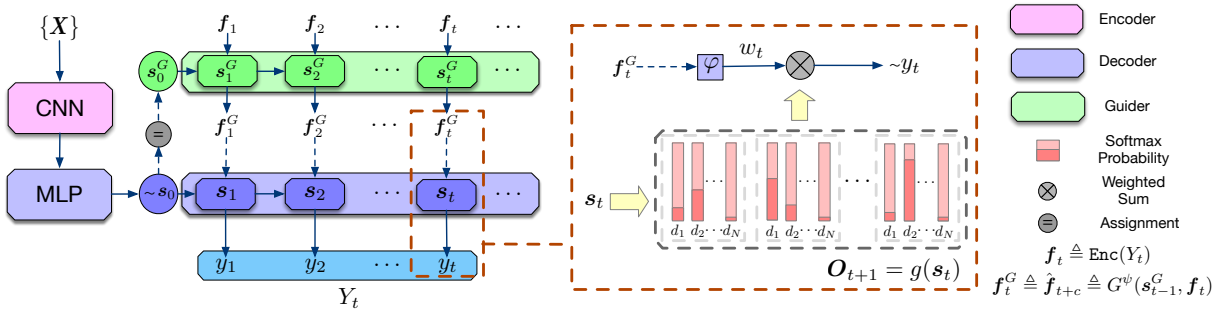

Figure 1: Model overview of text generation with a guider network. Solid lines mean gradients are backpropagated in training; dash lines mean gradients are not backpropagated. CNN is the feature extractor, and MLP outputs the parameters of the Gaussian density which is compatible with the initial state of the LSTM Guider and Decoder.

by an encoder network. Specifically, let the current generated sentence be $Y_{1...t}$ (encouraged to be the same as parts of a training sentence in training), with $\boldsymbol{f}_t$ calculated as: $\boldsymbol{f}_t = \text{Enc}(Y_{1...t})$. The initial state of the guider network is the encoded feature of a true input sentence by the same convolutional neural network (CNN), *i.e.*, $\boldsymbol{s}_0^G = \text{Enc}(\boldsymbol{X})$, where $\text{Enc}(\cdot)$ denotes the encoder transformation, implemented with a CNN (Zhang et al., 2017). Importantly, the input to the guider network, at each time point, is defined by features from the entire sentence generated to that point. This provides an important "guide" to the LSTM decoder, accounting for the global properties of the generated text.

Text Generation with Planning We first explain how one uses the guider network to guide next-word generation for the generator (the LSTM decoder in Figure 1). Our framework is inspired by the MPC method (Nagabandi et al., 2017), and can be regarded as a type of plan-ahead attention mechanism. Given the feature $\boldsymbol{f}_t$ at time t from the current input sentence, the guider network produces a prediction $G^\psi(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t)$ as a future feature representation, by feeding $\boldsymbol{f}_t$ into the LSTM guider. Since the training of the guider network is based on real data (detailed in the next paragraph), the predicted feature contains global-structure information of the training sentences. To utilize such information to predict the next word, we combine the predicted feature with the output of the decoder by constructing an attention-like mechanism. Specifically, we first apply a linear transformation φ on the predicted feature $G^\psi(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t)$, forming a weight vector $\boldsymbol{w}_t \triangleq \varphi\left(G^\psi(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t)\right)$. The weight $\boldsymbol{w}_t$ is applied to the output $\boldsymbol{O}_t$ of the LSTM decoder by an element-wise multiplication operation. The result is then fed into a softmax layer to generate the next token y_t. Formally, the generative process is written as:

$$\boldsymbol{O}_t = g(\boldsymbol{s}_{t-1}), \quad \boldsymbol{w}_t = \varphi(G^\psi(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t)), \tag{5}$$

$$y_t \sim \text{Multi}(1, \text{softmax}(\boldsymbol{O}_t \cdot \boldsymbol{w}_t)), \tag{6}$$

$$\boldsymbol{s}_t^G = h^G(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t), \quad \boldsymbol{s}_t = h(\boldsymbol{s}_{t-1}, e(y_t)) \ . \tag{7}$$

Guider Network Training Given a sentence of feature representations $(\boldsymbol{f}_1, \boldsymbol{f}_2, \dots \boldsymbol{f}_T)$ for a training sentence, we seek to update the guider network such that it is able to predict $\boldsymbol{f}_{t+c}$ given $\boldsymbol{f}_t$, where $c > 0$ is the number of steps that are looked ahead. We implement this by forcing the predicted feature, $G^\psi(\boldsymbol{s}_t^G, \boldsymbol{f}_t)$, to match both the sentence feature $\boldsymbol{f}_{t+c}$ (first term in (8)) and the corresponding feature-changing direction (second term in (8)). This is formalized by maximizing an objective function of the following form at time t:

$$J_G^\psi = \mathcal{D}_{\cos}\left(\boldsymbol{f}_{t+c}, \ G^\psi(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t)\right) + \mathcal{D}_{\cos}\left(\boldsymbol{f}_{t+c} - \boldsymbol{f}_t, \ G^\psi(\boldsymbol{s}_{t-1}^G, \boldsymbol{f}_t) - \boldsymbol{f}_t\right) \ , \tag{8}$$

where $\mathcal{D}_{\cos}(\cdot, \cdot)$ denotes the cosine similarity[2]. By maximizing (8), an ideal guider network should be able to predict the true next words conditioned on the current word in a sentence. As a result, the prediction is used to construct an intermediate reward, used to update the generator (the LSTM decoder), as described further below.

3.2 Feature-Matching Rewards and Generator Optimization

As in many RL-based text-generation methods, such as SeqGAN (Yu et al., 2017) and LeakGAN (Guo et al., 2017), the generator is updated based on policy-gradient methods. As a result, collecting rewards in the generation process is critical.

[2]We found that the cosine similarity worked better than the l_2-norm.

Though SeqGAN (Yu et al., 2017) has proposed to use rollout to get rewards for each generated word, the variance of the rewards is typically too high to be useful practically. In addition, the computational cost may be too high for practical use. We below describe how to use the proposed guider network to define intermediate rewards, leading to a definition of feature-matching reward.

Feature-Matching Rewards We first define an intermediate reward to generate a particular word. The idea is to match the ground-truth features from the CNN encoder in Figure 1 with those generated from the guider network. Equation (8) indicates that the further the generated feature is from the true feature, the smaller the reward should be. To this end, for each time t, we define the intermediate reward for generating the current word as:

$$r_t^g = \frac{1}{2c}\sum_{i=1}^{c}(\mathcal{D}_{cos}(\boldsymbol{f}_t, \hat{\boldsymbol{f}}_t) + \mathcal{D}_{cos}(\boldsymbol{f}_t - \boldsymbol{f}_{t-i}, \hat{\boldsymbol{f}}_t - \boldsymbol{f}_{t-i}))\ ,$$

where $\hat{\boldsymbol{f}}_t = G^{\psi}(s^G_{t-c-1}, \boldsymbol{f}_{t-c})$ is the predicted feature. Intuitively, $\boldsymbol{f}_t - \boldsymbol{f}_{t-i}$ measures the difference between the generated sentences in feature space; the reward is high if it matches the predicted feature transition $\hat{\boldsymbol{f}}_t - \boldsymbol{f}_{t-i}$ from the guider network. At the last step of text generation, *i.e.*, $t = T$, the corresponding reward measures the quality of the whole generated sentence, thus it is called a final reward. The final reward is defined differently from the intermediate reward, discussed below for both the unconditional- and conditional-generation cases.

Note that a token generated at time t will influence not only the rewards received at that time but also the rewards at subsequent time steps. Thus we propose to define the cumulative reward, $\sum_{i=t}^{T}\gamma^i r_i^g$ with γ a discount factor, as a *feature-matching reward*. Intuitively, this encourages the generator to focus on achieving higher long-term rewards. Finally, in order to apply policy gradient to update the generator, we combine the feature-matching reward with the problem-specific final reward, to form a Q-value reward specified below.

Similar to SeqGAN, the final reward is defined as the output of a discriminator, evaluating the quality of the whole generated sentence, *i.e.*, the smaller the output, the less likely the generation is a true sentence. As a result, we combine the adversarial reward $r^f \in [0, 1]$ by the discriminator (Yu et al., 2017) with the guider-matching rewards, to define a Q-value reward as $Q_t = (\sum_{i=t}^{T}\gamma^i r_i^g) \times r^f$.

Algorithm 1 Model-based Imitation Learning for Text Generation

Require: generator policy π^{ϕ}; guider network G^{ψ}; a sequence dataset $\{X_{1...T}\}$ by some expert policy.

1: Initialize G^{ψ}, D^{θ} with random weights.
2: **while** Imitation Learning phase **do**
3: Update generator π^{ϕ}, guider G^{ψ} with MLE loss.
4: **end while**
5: **while** Reinforcement Learning phase **do**
6: Generate a sequence $Y_{1...T} \sim \pi^{\phi}$.
7: Compute Q_t, and update π^{ϕ}.
8: **end while**

Generator Optimization The generator is initialized by pre-training on sentences with an autoencoder structure, based on MLE training. After that, the final Q-value reward Q_t is used as a reward for each time t, with standard policy gradient optimization methods to update the generator. Specifically, the policy gradient is

$$\nabla_{\phi} J = \mathbb{E}_{(\boldsymbol{s}_{t-1}, y_t)\sim\rho_{\pi}}\left[Q_t \nabla_{\phi} \log p(y_t|\boldsymbol{s}_{t-1}; \phi, \varphi)\right]\ ,$$
$$\nabla_{\varphi} J = \mathbb{E}_{(\boldsymbol{s}_{t-1}, y_t)\sim\rho_{\pi}}\left[Q_t \nabla_{\varphi} \log p(y_t|\boldsymbol{s}_{t-1}; \phi, \varphi)\right]\ ,$$

where $p(y_t|\boldsymbol{s}_{t-1}; \phi, \varphi)$ is the probability of generating y_t given $\boldsymbol{s}_{t-1}$ in the generator. Algorithm 1 describes the proposed model-based imitation learning framework for text generation.

Model-based or Model-free Text generation seeks to generate the next word (action) given the current (sub-)sentence (state). The generator is considered as an agent that learns a policy to predict the next word given its current state. In previous work (Ranzato et al., 2016), a metric reward is given and the generator is trained to only maximize the metric reward by trial, thus this is model-free learning. In the proposed method, the guider network models the environment dynamics, and is trained by minimizing the cosine similarity between the prediction and the ground truth on real text. For generator training, it maximizes the reward which is determined by the metric and guider network, and thus is model-free learning with model-based boosting (Gu et al., 2016). The model predictive control scheme is included in our method, where the guider network is used to help next-word selection at each time-step.

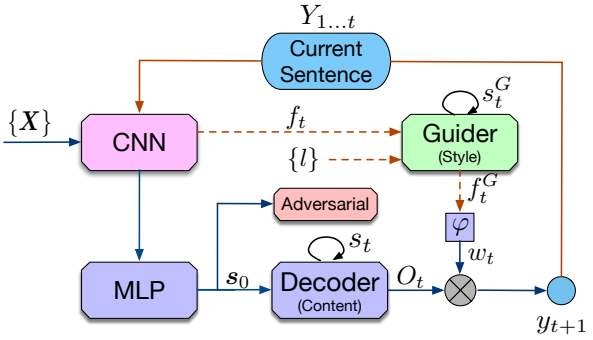

Figure 2: Guided style transfer: the Guider network controls the sentiment in the higher level, and the Generator focuses on preserving content in the lower level.

4 Extension to Non-parallel Text Style Transfer

As illustrated in Figure 2, our framework naturally provides a way for style transfer, where the guider network plays the role of style selection, and the generator only focuses on maintaining content without considering the styles. To make the guider network focus on the guidance of styles, we assign the label l as the initial state $\boldsymbol{s}_0^G$ of the guider network. Specifically, at each step t, we feed the current sentence representation $\boldsymbol{f}_t$ and label l into the guider network:

$$\boldsymbol{O}_t = g(\boldsymbol{s}_{t-1}), \quad \boldsymbol{w}_t = \varphi(G^{\psi}(\boldsymbol{s}_{t-1}^G, [\boldsymbol{f}_t, l])), \quad (9)$$

$$y_t \sim \text{Multi}(1, \text{softmax}(\boldsymbol{O}_t \cdot \boldsymbol{w}_t)). \quad (10)$$

For the generator, we put an adversarial regularizer on the encoded latent $\boldsymbol{s}_0(X)$ and penalize it if it contains the sentiment information, by maximizing the entropy, *i.e.,* $\max \sum_l p(l|\ \boldsymbol{s}_0(X)) \log p(l|\ \boldsymbol{s}_0(X))$, where p is a pre-trained classifier. Intuitively, the generator gives candidate words represented by O_t, while the guider makes a choice implicitly by w_t based on the sentiment information. The sentiment information is contained in w_t, while the content of the original sentence is represented by O_t. To achieve style-transfer, one feeds the original sentence X with the target style label l to get the transferred sentence Y with style l. Following previous work (Hu et al., 2017; Yang et al., 2018; Cheng et al., 2020), we adopt a classifier as the discriminator and the soft-argmax approach (Kusner and Miguel, 2016) for the update of generator instead of policy gradient (Sutton and Barto, 1998).

5 Related Work

We first review related works that combine RL and GAN for text generation. As one of the most representative models in this direction, SeqGAN (Yu et al., 2017) adopts Monte-Carlo search to calculate rewards. However, such a method introduces high variance in policy optimization. There are a number of works proposed subsequently to improve the reward-generation process. For example, RankGAN (Lin et al., 2017) proposes to replace the reward from the GAN discriminator with a ranking-based reward, MaliGAN (Che et al., 2017) modifies the GAN objective and proposes techniques to reduce gradient variance, MaskGAN (Fedus et al., 2018) uses a filling technique to define a Q-value reward for sentence completion, RelGAN (Nie et al., 2019) uses a relational memory based generator for the long-distance dependency modeling, FM-GAN (Chen et al., 2018) uses a feature mover distance to match features of real and generated sentences inspired by optimal transport (Chen et al., 2019; Zhang et al., 2018), and LeakGAN (Guo et al., 2017) tries to address the sparse-reward issue for long-text generation with hierarchical RL by utilizing the leaked information from a GAN discriminator. One problem of LeakGAN is that it tends to overfit the training data, yielding generated sentences that are often not diverse. By contrast, by relying on a model-based imitation learning approach, our method learns global-structure information, which generates more-diverse sentences, and can be extended to conditional text generation. Zhang et al. (2020) designed a differentiable nested Wasserstein distance for semantic matching, which can be applied for further improvement.

RL techniques can also be used in other ways for text generation (Bachman and Precup, 2015). For example, Ranzato et al. (2016) trained a Seq2Seq model by directly optimizing the BLEU/ROUGE scores with the REINFORCE algorithm. To reduce variance of the vanilla REINFORCE, Bahdanau et al. (2017) adopted the actor-critic framework for sequence prediction. Furthermore, Rennie et al. (2016) trained a baseline algorithm with a greedy decoding scheme for the REINFORCE method. Note that all these methods can only obtain reward after a whole sentence is generated. Planning techniques in RL have also been explored to improve text generation (Gulcehre et al., 2017; Serdyuk et al., 2018). Zhang et al. (2020) introduced the self-imitation scheme to exploit historical high-quality sentences for enhanced exploration. Compared to these related works, the proposed guider network can provide a planning mechanism and intermediate rewards.

Method	Test-BLEU-2	3	4	5	Self-BLEU-2	3	4
SeqGAN (Yu et al., 2017)	0.820	0.604	0.361	0.211	0.807	0.577	0.278
RankGAN (Lin et al., 2017)	0.852	0.637	0.389	0.248	0.822	0.592	0.230
GSGAN (Kusner and Miguel, 2016)	0.810	0.566	0.335	0.197	0.785	0.522	0.230
TextGAN (Zhang et al., 2017)	0.910	0.728	0.484	0.306	0.806	0.548	0.217
LeakGAN (Guo et al., 2017)	0.922	0.797	0.602	0.416	0.912	0.825	0.689
MLE (Caccia et al., 2018)	0.902	0.706	0.470	0.392	0.787	0.646	0.485
GMGAN (ours)	0.949	0.823	0.635	0.421	0.746	0.511	0.319

Table 1: Test-BLEU (↑) and Self-BLEU (↓) scores on Image COCO.

Method	Test-BLEU-2	3	4	5	Self-BLEU-2	3	4
SeqGAN (Yu et al., 2017)	0.630	0.354	0.164	0.087	0.728	0.411	0.139
RankGAN (Lin et al., 2017)	0.723	0.440	0.210	0.107	0.672	0.346	0.119
GSGAN (Kusner and Miguel, 2016)	0.723	0.440	0.210	0.107	0.807	0.680	0.450
TextGAN (Zhang et al., 2017)	0.777	0.529	0.305	0.161	0.806	0.662	0.448
LeakGAN (Guo et al., 2017)	0.923	0.757	0.546	0.335	0.837	0.683	0.513
MLE (Caccia et al., 2018)	0.902	0.706	0.470	0.392	0.787	0.646	0.485
GMGAN (ours)	0.923	0.727	0.491	0.303	0.814	0.576	0.328

Table 2: Test-BLEU (↑) and Self-BLEU (↓) scores on EMNLP2017 WMT News.

6 Experiments

We test the proposed framework on unconditional and conditional text generation tasks, and analyze the results to understand the performance gained by the guider network. We also perform an ablation investigation on the improvements brought by each part of our proposed method, and consider non-parallel style transfer. All experiments are conducted on a single Tesla P100 GPU and implemented with TensorFlow and Theano. Details of the datasets, the experimental setup and model architectures are provided in the Appendix.

6.1 Implementation Details

Encoder as the feature extractor For unconditional generation, the feature extractor that generates inputs for the guider network shares the CNN part of the encoder. We stop gradients from the guider network to the encoder CNN in the training process. For conditional generation, we use a pre-trained feature extractor, trained similarly to the unconditional generation.

Training procedure As with many imitation-learning models (Bahdanau et al., 2017; Rennie et al., 2016; Sutskever et al., 2014), we first train the encoder-decoder part based on the off-policy data with an MLE loss. Then we use RL training to fine-tune the trained generator. We adaptively transfer the training from MLE loss to RL loss, similar to (Paulus et al., 2017; Ranzato et al., 2016).

Initial states We use the same initial state for both the generator and the guider networks. For conditional generation, the initial state is the encoded latent code of the conditional information for both training and testing. For unconditional generation, the initial state is the encoded latent code of a target sentence in training and random noise in testing.

6.2 Adversarial Text Generation

We focus on adversarial text generation, and compare our approach with a number of related works (Guo et al., 2017; Lin et al., 2017; Yu et al., 2017; Zhang et al., 2017; Zhu et al., 2018). In this setting, a discriminator in the GAN framework is added to the model in Figure 1 to guide the generator to generate high-quality sentences. This is implemented by defining the final reward to be the output of the discriminator. All baseline experiments are implemented on the texygen platform (Zhu et al., 2018). We adopt the BLEU score, referenced by the test set (test-BLEU, higher value implies better quality) and itself (self-BLEU, lower value implies better diversity) (Zhu et al., 2018) to evaluate quality of generated samples, where test-BLEU evaluates the reality of generated samples, and self-BLEU measures the diversity. A good generator should achieve both a high test-BLEU score and a low self-BLEU score. In practice, we use $\triangle t = c = 4$ and $\gamma = 0.25$. We call the proposed method guider-matching GAN (GMGAN) for unconditional text

generation. More details of GMGAN are provided in Appendix D.

Short Text Generation: COCO Image Captions We use the COCO Image Captions Dataset, in which most sentences have a length of about 10 words. Since we consider unconditional text generation, only image captions are used as the training data. After preprocessing, we use 120,000 random sample sentences as the training set, and 10,000 as the test set. The BLEU scores with different methods are listed in Table 1. We observe that GMGAN performs significantly better than the baseline models. Specifically, besides achieving higher test-BLEU scores, the proposed method also generates samples with very good diversity in terms of self-BLEU scores. LeakGAN represents the state-of-the-art in adversarial text generation, however, its diversity measurement is relatively poor (Zhu et al., 2018). We suspect that the high BLEU score achieved by LeakGAN is due to its mode collapse on some good samples, resulting in high self-BLEU scores. Other baselines achieve lower self-BLEU scores since they cannot generate reasonable sentences.

Long Text Generation: EMNLP2017 WMT Following (Zhu et al., 2018), we use the News section in the EMNLP2017 WMT4 Dataset as our training data. The dataset consists of 646,459 words and 397,726 sentences. After preprocessing, the training dataset contains 5,728 words and 278,686 sentences. The BLEU scores with different methods are provided in Table 2. Compared with other methods, LeakGAN and GMGAN achieve comparable test-BLEU scores, demonstrating high-quality generated sentences. Again, LeakGAN tends to over-fit on training data, leading to much higher (worse) self-BLEU scores. Our proposed GMGAN shows good diversity of long text generation with lower self-BLEU scores. Other baselines obtain both low self-BLEU and test-BLEU scores, leading to more random generations.

Human Evaluation Simply relying on the above metrics is not sufficient to evaluate the proposed method (Caccia et al., 2018). Following previous work (Guo et al., 2017), we perform human evaluations using Amazon Mechnical Turk, evaluating the text quality based on readability and meaningfulness (whether sentences make sense) on the EMNNLP2017 WMT News dataset. We ask the worker to rate the input sentence with scores scaling from 1 to 5, with 1 as the worst score and 5 as the best. The detailed criteria is listed in Table 3. We require all the workers to be native English speakers, with approval rate higher than 90% and at least 100 assignments completed.

Scores	Criteria
5 (Best)	It is consistent, informative, grammatically correct.
4	It is grammatically correct and makes sense.
3	It is mostly meaningful and with small grammatical error.
2	It needs some time to understand and has grammatical errors.
1 (Worst)	Meaningless, not readable.

Table 3: Human evaluation rating criteria.

Methods	MLE	SeqGAN	RankGAN	GSGAN
Human scores	2.45±0.14	2.57±0.15	2.91±0.17	2.48±0.14
Methods	**textGAN**	**LeakGAN**	**GMGAN**	**Real**
Human scores	3.11±0.16	3.47±0.15	3.89±0.15	4.21±0.14

Table 4: Results of human evaluation with different methods on EMNLP2017 WMT dataset.

We randomly sample 100 sentences generated by each model. Ten native English speakers on Amazon Mechanical Turk are asked to rate each sentence. The average human rating scores are shown in Table 4, indicating GMGAN achieves higher human scores compared to other methods. As examples, Table 5 illustrates some generated samples by GMGAN and its baselines. The performance on the two datasets indicates that the generated sentences of GMGAN are of higher global consistency and better readability than SeqGAN and LeakGAN. More generated examples are provided in the Appendix.

Ablation Study We conduct ablation studies on long text generation to investigate the improvements brought by each part of our proposed method. We first test the benefits of using the guider network. Among the methods compared, Guider is the standard MLE model with the guider network. We further compare RL training with *i*) only final rewards , *ii*) only feature-matching rewards, and *iii*) combining both rewards, namely GMGAN. The results are shown in Table 6. We observe that guider network plays an important role in improving the performance. RL training with final rewards given by a discriminator typically damages the generation quality, but feature-matching reward produces sentences with much better diversity due to the ability of exploration.

Method	COCO Image Captions	EMNLP2017 WMT News
SeqGAN	(1) A person and black wooden table. (2) A closeup of a window at night.	(1) She added on a page where it was made clear more old but public got said. (2) I think she're guys in four years , and more after it played well enough.
LeakGAN	(1) A bathroom with a black sink and a white toilet next to a tub. (2) A man throws a Frisbee across the grass covered yard.	(1)"I'm a fan of all the game, I think if that's something that I've not," she said, adding that he would not be decided. (2) The UK is Google' s largest non-US market, he has added "20, before the best team is amount of fewer than one or the closest home or two years ago.
GMGAN	(1) Bicycles are parked near a row of large trees near a sidewalk. (2) A married couple posing in front of a piece of birthday cake.	(1) "Sometimes decisions are big, but they're easy to make," he told The Sunday Times in the New Year. (2) A BBC star has been questioned by police on suspicion of sexual assault against a 23-year-old man , it was reported last night.

Table 5: Examples of generated samples with different methods on COCO and EMNLP datasets.

Methods	MLE	Guider	Final	Stepwise	GMGAN
Test-BLEU-2	0.761	0.920	0.843	0.914	0.923
BLEU-3	0.468	0.723	0.623	0.704	0.727
BLEU-4	0.230	0.489	0.390	0.457	0.491
BLEU-5	0.116	0.289	0.221	0.276	0.303
Self-BLEU-2	0.664	0.812	0.778	0.798	0.814
BLEU-3	0.338	0.589	0.525	0.563	0.576
BLEU-4	0.113	0.360	0.273	0.331	0.328

Table 6: Ablation study on EMNLP2017 WMT.

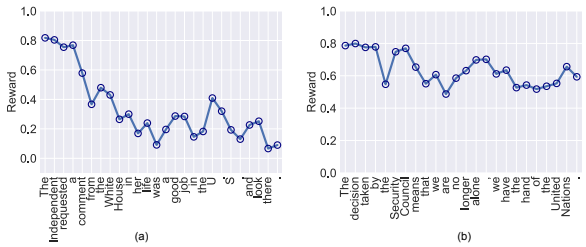

Figure 3: Guider-Matching Rewards Illustrations.

Case Study of Guider-Matching Rewards Figure 3(a) illustrates the feature-matching rewards in the generation. Figure 3(a) shows an example of failure generation in the training stage, when two sentences are combined by the word '*was*'. It is grammatically wrong to select '*was*' for the generator, thus the guider network gives a small reward. We can see that the rewards become lower with more time steps, which is consistent with the exposure bias. Figure 3(b) shows a successful generation, where the rewards given by the guider are relatively high (larger than 0.5). These observations validate that: (*i*) exposure bias exists in MLE training. (*ii*) RL training with exploration can help reduce the effects of exposure bias. (*iii*) Our proposed feature-matching rewards can provide meaningful guidance to maintain sentence structure and fluency.

Model	Acc(%)	BLEU	BLEU-ref
CVAE (Shen et al., 2017)	73.9	20.7	7.8
Controllable (Hu et al., 2017)	86.7	58.4	-
BackTrans (Prabhumoye et al., 2018)	91.2	2.8	2.0
DeleteAndRetrieval (Li et al., 2018a)	88.9	36.8	14.7
Guider (Ours)	**92.7**	**52.1**	**25.4**

Table 7: Non-parallel text style transfer results on the test set with *human* references.

6.3 Non-parallel Text-style Transfer

We test the proposed framework on the non-parallel text-style-transfer task, where the goal is to transfer one sentence in one style (*e.g.*, positive) to a similar sentence but with a different style (*e.g.*, negative). Pair-wise information should be inferred from the training data, which becomes more challenging. For a fair comparison, we use the same data and its split method as in (Shen et al., 2017). Specifically, there are 444,000, 63,500, and 127,000 sentences with either positive or negative sentiments in the training, validation and test sets, respectively.

To measure whether the original sentences (in the test set) have been transferred to the desired sentiment, we follow the settings of (Shen et al., 2017) and employ a pretrained CNN classifier, which achieves an accuracy of 97.4% on the validation set, to evaluate the transferred sentences. We also report the BLEU scores with original sentences (BLEU) and human references (BLEU-ref) (Li et al., 2018a), to evaluate the content preservation of transferred sentences. Results are summarized in Table 7. Our proposed model exhibits higher transfer accuracy and better content preservation, indicating the guider network provides good sentiment guidance to better preserve the content information.

From positive to negative
Original: all the employees are **friendly** and **helpful** .
Transferred: all the employees are **rude** and **unfriendly** .

Original: i 'm so **lucky** to have found this place !
Transferred: i 'm so **embarrassed** that i picked this place .

From negative to positive
Original: the service was **slow** .
Transferred: the service was **fast** and **friendly** .

Original: i would **never** eat there again and would probably **not stay** there either .
Transferred: i would **definitely** eat this place and i would **recommend** them .

Table 8: Generated samples of guided style transfer.

7 Conclusions

We have proposed a model-based imitation-learning framework for adversarial text generation, by introducing a guider network to model the generation environment. The guider network provides a plan-ahead mechanism for next-word selection. Furthermore, this framework can alleviate the sparse-reward issue, as the intermediate rewards are used to optimize the generator. Our proposed models are validated on both unconditional and conditional text generation, including adversarial text generation and non-parallel style transfer. We achieve improved performance in terms of generation quality and diversity for unconditional and conditional generation tasks.

Acknowledgement The authors would like to thank the anonymous reviewers for their insightful comments. The research was supported in part by DARPA, DOE, NIH, NSF and ONR.

References

Philip Bachman and Doina Precup. 2015. Data generation as sequential decision making. In *NIPS*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *ICLR*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*.

Nir Baram, Oron Anschel, and Shie Mannor. 2017. Model-based adversarial imitation learning. In *ICML*.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2018. Language gans falling short. *arXiv:1811.02549*.

Tong Che, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. In *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 *NeurIPS*.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving sequence-to-sequence learning via optimal transport. In *ICLR*.

Ching-An Cheng, Xinyan Yan, Evangelos Theodorou, and Byron Boots. 2019. Accelerating imitation learning with predictive models. In *AISTATS*.

Pengyu Cheng, Renqiang Min, Dinghan Shen, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information theoretical guidance. In *ACL*.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better text generation via filling in the _. *ICLR*.

Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous deep q-learning with model-based acceleration. In *ICML*.

Caglar Gulcehre, Francis Dutil, Adam Trischler, and Yoshua Bengio. 2017. Plan, attend, generate: Character-level neural machine translation with planning. In *NIPS*.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Long text generation via adversarial training with leaked information. In *AAAI*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Controllable text generation. In *ICML*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. 2018. Model-ensemble trust-region policy optimization. In *ICLR Workshop*.

Matt J Kusner and Hernández-Lobato José Miguel. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018a. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *NAACL*.

Piji Li, Lidong Bing, and Wai Lam. 2018b. Actor-critic based training framework for abstractive summarization. In *arXiv:1803.11070*.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *NIPS*.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. 2017. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *ICRA*.

Weili Nie, Nina Narodytska, and Ankit Patel. 2019. Relgan: Relational generative adversarial networks for text generation. In *ICLR*.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *ICML*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. In *ICLR*.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *ACL*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. In *CVPR*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv:1509.00685*.

Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordoni, Adam Trischler, Chris Pal, and Yoshua Bengio. 2018. Twin networks: Matching the future for sequence generation. In *ICLR*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. MIT Press.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Ruiyi Zhang, Changyou Chen, Zhe Gan, Zheng Wen, Wenlin Wang, and Lawrence Carin. 2020. Nested-wasserstein self-imitation learning for sequence generation. In *AISTATS*.

Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. 2018. Policy optimization as wasserstein gradient flows. In *ICML*.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *ICML*.

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 *SIGIR*.

A Additional Experiments

More Generated Samples of Text Generation Table 13 lists more generated samples on the proposed GMGAN and its baselines. From the experiments, we can see, (i) SeqGAN tends to generate shorter sentences, and the readability and fluency is very poor. (ii) LeakGAN tends to generate very long sentences, and usually longer than the original sentences. However, even with good locality fluency, its sentences usually are not semantically consistent. By contrast, our proposed GMGAN can generate sentences with similar length to the original sentences, and has good readability and fluency. This is also validated in the Human evaluation experiment.

Method	BLEU-3	BLEU-4	METEOR	CIDEr
No attention, Greedy, Resnet-152				
MLE	37.2	26.5	23.1	83.9
Guider	38.0	27.3	23.9	85.4
MIXER (BLEU)	39.1	29.3	22.3	79.7
SCST (BLEU)	41.6	31.6	23.1	87.5
GMST (BLEU)	**41.8**	**32.1**	23.4	87.9
MIXER (CIDEr)	39.1	27.7	23.0	90.9
SCST (CIDEr)	41.2	30.0	24.3	98.6
GMST (CIDEr)	41.3	30.3	**24.4**	**100.1**
No attention, Greedy, Tag				
MLE	39.4	28.8	24.4	91.3
Guider	39.6	29.0	24.6	92.7
MIXER (BLEU)	42.4	32.2	23.7	90.4
SCST (BLEU)	43.9	33.6	24.5	95.9
GMST (BLEU)	**44.3**	**33.9**	24.5	97.1
MIXER (CIDEr)	42.1	30.8	24.7	101.2
SCST (CIDEr)	43.6	32.1	25.4	105.5
GMST (CIDEr)	44.1	32.6	**25.5**	**107.4**

Table 9: Results for image captioning on the MS COCO dataset; the higher the better for all metrics.

Image Captioning We conduct experiments on image captioning (Karpathy and Fei-Fei, 2015), investigating benefits brought by the Guider network. In image captioning, instead of using a discriminator to define final rewards for generated sentence, we adopt evaluation metrics computed based on human references. The final rewards appear more important as they contain reference (ground-truth) information. Feature-matching rewards work as a regularizer of the final rewards. We call our model in this setting a guider-matching sequence training (GMST) model. An overview of GMST is provided in the Appendix. We test our proposed model on the MS COCO dataset (Karpathy and Fei-Fei, 2015), containing 123,287 images in total. Each image is annotated with at least 5 captions. Following Karpathy's split (Karpathy and Fei-Fei, 2015), 5,000 images are used for both validation and testing. We report BLEU-k (k from 1 to 4), CIDEr (Vedantam et al., 2015), and METEOR (Banerjee and Lavie, 2005) scores. We consider two settings: (*i*) using a pre-trained 152-layer ResNet (He et al., 2016) for feature extraction, where we take the output of the 2048-way *pool5* layer from ResNet-152, pretrained on the ImageNet dataset; and (*ii*) using semantic tags detected from the image as features (Gan et al., 2017). We use an LSTM with 512 hidden units with mini-batches of size 64. Adam (Kingma and Ba, 2014) is used for optimization, with learning rate 2×10^{-4}. We pretrain the captioning model for the maximum 20 epochs, then use the reinforcement learning to train it for 20 epochs and test on the best model on the validation set.

The results are summarized in Table 9. When comparing an AutoEncoder (AE) with a variant implemented by adding a guider network (Guider), improvements are observed. We compare the proposed GMST with SCST. Note the main difference between GMST and SCST is that the former employs our proposed feature-matching reward, while the latter only considers the final reward provided by evaluation metrics. GMST achieves higher scores compared with SCST on its optimized metrics. The gain of GMST compared with SCST comes from the immediate rewards, which can maintain the semantic consistency and sentence structure, preventing language-fluency damage caused by only focusing on evaluation metrics. Specifically, the average length of generated sentence with a Guider is 15.7, and 12.9 for traditional generator.

Comparison with MLE The guider network models the long-term dependency and overcome the issue of sparse reward inspired by model predictive control (MPC). The experiments aim to quantify the gain when incorporating MPC for imitation learning, i.e., MLE and RL finetune.

We provide an additional comparison with Caccia et al. (2018) and evaluate the diversity and quality with BLEU scores. We also report the F1-BLEU which considers both diversity and quality in Table 10.

B Discussions of the Guider Network

Guider network can be regarded as a model of the text-generation environments, namely the model of dynamics. It takes current $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$ as input, and outputing an estimate of the next state $\boldsymbol{s}_{t+\triangle t}$

Method	Test-BLEU-2	3	4	Self-BLEU-2	3	4	F1-BLEU-2	3	4
MLE (Caccia et al., 2018)	0.902	0.706	0.470	0.787	0.646	0.485	**0.345**	0.472	0.491
Guider (MLE)	0.920	0.723	0.489	0.812	0.589	0.360	0.312	0.524	0.554
GMGAN (Ours)	0.923	0.727	0.491	0.814	0.576	0.328	0.310	**0.537**	**0.567**

Table 10: Additional Comparison with MLE (Caccia et al., 2018) .

at time $t + \triangle t$. In the text generation setting, when $\triangle t = 1$, we can exactly get the feature representation of the current generated sentence if the guider does not help the word selection. If not, we cannot exactly get this feature extraction since the guider's prediction partly determine next token. In practice, we use $\triangle t = c = 4$, to give the guider planning ability, to help for word selection and guide sentence generation.

C Experimental Setup

C.1 Adversarial Text Generation

For Image COCO, the learning rate of the generator is 0.0002, the learning rate of the guider 0.0002, the maximum length of sequence is 25. For WMT, the learning rate of the guider 0.0002, the learning rate of the guider 0.0002, the maximum length of sequence is 50. We use $c = 4$ chosen from $[2, 3, 4, 5, 8]$ and $\gamma = 0.25$ chosen from $[0.1, 0.25, 0.5, 0.75, 0.99]$. We use Adam (Kingma and Ba, 2014) optimization algorithm to train the guider, generator and discriminator.

For both tasks, the LSTM state of dimension for the generator is 300, and the LSTM state of dimension for the generator is 300. The dimension of word-embedding is 300. The output dimension of the linear transformation connecting guider and generator is 600×10. The learning rate of Discriminator is 0.001.

C.2 Conditional Generation

For Image Captioning, the learning rate of the guider 0.0002, the learning rate of the guider 0.0002, the maximum length of sequence is 25. For Style transfer, the learning rate of the guider 0.0001, the learning rate of the guider 0.0001, the maximum length of sequence is 15.

C.3 Network Structure of Models

The LSTM state of dimension for the generator is 300, and the LSTM state of dimension for the guider is 300. The dimension of word-embedding is 300.

(Sub-)sequence to latent features
Input 300× Seq. Length Sequences
5×300 conv. 300 ReLU, stride 2 5×1 conv. 600 ReLU, stride 2 MLP output 600, ReLU

Table 11: Architecture of Encoder.

Sequence to a scalar value
Input 300× Seq. Length Sequences
5×300 conv. 300 ReLU, stride 2 5×1 conv. 600 ReLU, stride 2 MLP output 1, ReLU

Table 12: Architecture of Discriminator.

D Algorithm Details

Algorithm 2 Guider Matching Generative Adversarial Network (GMGAN)

Require: generator policy π^ϕ; discriminator D_θ; guider network G^ψ; a sequence dataset $\mathcal{S} = \{X_{1...T}\}$.

Initialize G^ψ, π^ϕ, D^θ with random weights.
Pretrain generator π^ϕ, guider G^ψ and discriminator D^θ with MLE loss.
repeat
  for g-steps **do**
    Generate a sequence $Y_{1...T} \sim \pi^\phi$.
    Compute Q_t via (5), and update π^ϕ with policy gradient via (8).
  end for
  for d-steps **do**
    Generate a sequences from π^ϕ.
    Train discriminator D_θ.
  end for
until GMGAN converges

	**Res152-SCST:** a group of zebras standing in a field . **Res152-GMST:** a herd of zebras standing in a field of grass . **Tag-SCST:** a zebra and a zebra drinking water from a field of grass . **Tag-GMST:** a group of zebras drinking water in the field of grass .		**Res152-SCST:** a group of people walking down a skateboard . **Res152-GMST:** a group of people standing on a street with a skateboard . **Tag-SCST:** a woman walking down a street with a skateboard . **Tag-GMST:** a black and white photo of a man riding a skateboard .
	**Res152-SCST:** a baby sitting next to a baby giraffe . **Res152-GMST:** a little baby sitting next to a baby holding a teddy bear . **Tag-SCST:** a black and white photo of a woman holding a teddy bear . **Tag-GMST:** a black and white photo of a man and a woman holding a teddy bear .		**Res152-SCST:** a traffic light on a street with a in the . **Res152-GMST:** a traffic light on the side of a street . **Tag-SCST:** a traffic light on a street with a green . **Tag-GMST:** a red traffic light sitting on the side of a road .

Figure 4: Examples of image captioning on MS COCO.

Algorithm 3 Guider Matching Sequence Training (GMST)

Require: generator policy π^{ϕ}; discriminator D_{θ}; guider network G^{ψ}; a sequence dataset $\mathcal{S} = \{Y_{1...T}\}$ and its condition information $\mathcal{I} = \{X\}$

1: Initialize G^{ψ}, π^{ϕ}, D^{θ} with random weights.
2: Pretrain generator π^{ϕ}, guider G^{ψ} and discriminator D^{θ} with MLE loss.
3: **repeat**
4: Generate a sequence $Y_{1...T} \sim \pi^{\phi}$.
5: Compute evaluation scores based on references.
6: Compute Q_t^s via (6), and update π^{ϕ} with policy gradient via (8).
7: **until** GMST converges

Method	Generated Examples
Real Data	What this group does is to take down various different websites it believes to be criminal and leading to terrorist acts .
	Over 1 , 600 a day have reached Greece this month , a higher rate than last July when the crisis was already in full swing .
	" We ' re working through a legacy period , with legacy products that are 10 or 20 years old ," he says .
	' The first time anyone says you need help , I ' m on the defensive , but that ' s all that I know .
	Out of those who came last year , 69 per cent were men , 18 per cent were children and just 13 per cent were women .
	He has not played for Tottenham ' s first team since and it is now nearly two years since he completed a full Premier League match for the club .
	So you have this man who seems to represent this way to live and how to be a good citizen of the world .
	CNN : You made that promise , but it wasn ' t until 45 years later that you acted on it .
	This is a part of the population that is notorious for its lack of interest in actually showing up when the political process takes place .
	They picked him off three times and kept him out of the end zone in a 22 - 6 victory at Arizona in 2013 .
	The treatment was going to cost £ 12 , 000 , but it was worth it for the chance to be a mum .
	But if black political power is so important , why hasn ' t it made more of a difference in the lives of poor black people in Baltimore such as Gray ?
	Local media reported the group were not looking to hurt anybody , but they would not rule out violence if police tried to remove them .
	The idea was that couples got six months ' leave per child with each parent entitled to half the days each .
	The 55 to 43 vote was largely split down party lines and fell short of the 60 votes needed for the bill to advance .
	Taiwan ' s Defence Ministry said it was " aware of the information ," and declined further immediate comment , Reuters reported .
	I ' m racing against a guy who I lost a medal to - but am I ever going to get that medal back ?
	Others pushed back their trips , meaning flights early this week are likely to be even more packed than usual .
	" In theory there ' s a lot to like ," Clinton said , " but ' in theory ' isn ' t enough .
	If he makes it to the next election he ' ll lose , but the other three would have lost just as much .
SeqGAN	Following the few other research and asked for " based on the store to protect older , nor this .
	But there , nor believe that it has reached a the person to know what never - he needed .
	The trump administration later felt the alarm was a their doctors are given .
	We have been the time of single things what people do not need to get careful with too hurt after wells then .
	If he was waited same out the group of fewer friends a more injured work under it .
	It will access like the going on an " go back there and believe .
	Premier as well as color looking to put back on a his is .
	So , even though : " don ' t want to understand it at an opportunity for our work .
	I was shocked , nor don ' t know if mate , don ' t have survived ,
	So one point like ten years old , but a sure , nor with myself more people substantial .
	And if an way of shoes of crimes the processes need to run the billionaire .
	Now that their people had trained and people the children live an actor , nor what trump had .
	However , heavily she been told at about four during an innocent person .
LeakGAN	The country has a reputation for cheap medical costs and high - attack on a oil for more than to higher its - wage increase to increase access to the UK the UK women from the UK ' s third nuclear in the last couple of weeks .
	I ' ve been watching it through , and when the most important time it is going to be so important .
	I ' m hopeful that as that process moves along , that the U . S . Attorney will share as much as far as possible .
	The main thing for should go in with the new contract , so the rest of the Premier League is there to grow up and be there ," she said .
	I think the main reason for their sudden is however , I didn ' t get any big thing ," he says , who is the whole problem on the U . S . Supreme Court and rule had any broken .
	The average age of Saudi citizens is still very potential for the next year in the past year , over the last year he realised he has had his massive and family and home .
	" I think Ted is under a lot of people really want a " and then the opportunity to put on life for security for them to try and keep up .
	The new website , set to launch March 1 , but the U . S is to give up the time the case can lead to a more than three months of three months to be new home .
	It ' s a pub ; though it was going to be that , but , not , but I am not the right thing to live ," she said .
	" I ' m not saying method writing is the only way to get in the bedroom to get through the season and we ' ll be over again ," he says .
	I ' m not suggesting that our jobs or our love our years because I have a couple of games where I want it to be .
	The German government said 31 suspects were briefly detained for questioning after the New Year ' s Eve trouble , among them not allowed to stay in the long - term .
	It was a punishment carried out by experts in violence , and it was hard to me he loved the man and he ' s got off to support me in the future .
	" I ' ve known him , all that just over the last two weeks and for the last 10 years , I ' ll have one day of my life ," she said .
	The main idea behind my health and I think we saw in work of our country was in big fourth - up come up with a little you ' ve ever .
	he Kings had needed scoring from the left side , too , and King has provided that since his return are the of the first three quarters of the game .
	It ' s going to be a good test for us and we are on the right way to be able to get through it on every day on the year .
GMGAN	But it ' s grown up a little now , and might be ready for actually putting into your house .
	More than a dozen Republicans and a handful of Democrats have announced they are running for their party ' s 2016 presidential nomination , and when they were wealthy in 2010 right , what he has .
	And with a growing following of more than 45 , 000 people on Facebook , awareness of their work is on the rise .
	In all age groups , for instance , more people cited retirement as the reason for being out of the labour force , and it wasn ' t a problem in big .
	I had to train really , really hard and that ' s the advice I can give , because if you don ' t work hard somebody else will .
	I am picking up two cars tomorrow and taking them down south tomorrow if all goes according to plan ," he said .
	The team looked into the influence of marriage on weight loss after surgery - as well as the effects of surgery on the quality of his administration and rest on the world .
	Two former prime ministers were set to face off in the second round of a presidential election in New Hampshire .
	A third more complaints were made about the accounts between April and December last year than in the whole of 2014 / 15 .
	United Airlines subsequently worked to get those passengers back in the air so they could get to Colorado , the airline spokesman said .
	Mr Brown was standing in the kitchen when he started to feel a bit cold - and he noticed the door had disappeared .
	She has focused instead on where she parts ways with her rival on other issues , like to have someone with a president has revealed .
	Once , an ex - boyfriend and I lived with her for two months after we came back from travelling .
	He had faced 10 years in prison on the charges but the first government have been made at the recent peak .
	" We weren ' t exposed to things we didn ' t have in the same way kids these days are ," said Obama .
	I have no idea what it is , but there is definitely an intelligence - a higher intelligence - at work you have you want to make sure you are going into the local community .
	His current club have confirmed they would be willing to listen to offers for the attacking midfielder , but we did not have the right manager - there ' s summer to be in a big .
	We are in the last 16 and the target is always to win in the Champions League and will continue at the best level to be the coach .
	People are seeing that you can go into real estate and do really well and do something we want and if we make the right decision , and how we will be doing it is .

Table 13: Generated Examples on EMNLP2017 WMT.

Original:	i 'm so lucky to have found this place !
Guider:	i 'm so embarrassed that i picked this place .
Original:	awesome place , very friendly staff and the food is great !
Guider:	disgusting place , horrible staff and extremely rude customer service .
Original:	this was my first time trying thai food and the waitress was amazing !
Guider:	this was my first experience with the restaurant and we were absolutely disappointed .
Original:	thanks to this place !
Guider:	sorry but this place is horrible .
Original:	the staff was warm and friendly .
Guider:	the staff was slow and rude .
Original:	great place and huge store .
Guider:	horrible place like ass screw .
Original:	the service is friendly and quick especially if you sit in the bar .
Guider:	the customer service is like ok - definitely a reason for never go back ..
Original:	everything is always delicious and the staff is wonderful .
Guider:	everything is always awful and their service is amazing .
Original:	best place to have lunch and or dinner .
Guider:	worst place i have ever eaten .
Original:	best restaurant in the world !
Guider:	worst dining experience ever !
Original:	you 'll be back !
Guider:	you 're very disappointed !
Original:	you will be well cared for here !
Guider:	you will not be back to spend your money .
Original:	they were delicious !
Guider:	they were overcooked .
Original:	seriously the best service i 've ever had .
Guider:	seriously the worst service i 've ever experienced .
Original:	it 's delicious !
Guider:	it 's awful .

Table 14: Sentiment transfer samples on Yelp dataset (positive → negative).

Original:	gross !
Guider:	amazing !
Original:	the place is worn out .
Guider:	the place is wonderful .
Original:	very bland taste .
Guider:	very fresh .
Original:	terrible service !
Guider:	great customer service !
Original:	this place totally sucks .
Guider:	this place is phenomenal .
Original:	this was bad experience from the start .
Guider:	the food here was amazing good .
Original:	very rude lady for testing my integrity .
Guider:	very nice atmosphere for an amazing lunch !
Original:	they recently renovated rooms but should have renovated management and staff .
Guider:	great management and the staff is friendly and helpful .
Original:	this store is not a good example of sprint customer service though .
Guider:	this store is always good , consistent and they 're friendly .
Original:	one of my least favorite ross locations .
Guider:	one of my favorite spots .
Original:	horrible in attentive staff .
Guider:	great front desk staff !
Original:	the dining area looked like a hotel meeting room .
Guider:	the dining area is nice and cool .
Original:	never ever try to sell your car at co part !
Guider:	highly recommend to everyone and recommend this spot for me !
Original:	i ordered the filet mignon and it was not impressive at all .
Guider:	i had the lamb and it was so good .

Table 15: Sentiment transfer samples on Yelp dataset (negative $\rightarrow$ positive).