arXiv:2004.14507v1 [cs.LG] 29 Apr 2020

# Counterfactual Off-Policy Training for Neural Response Generation

**Qingfu Zhu**[♯], **Weinan Zhang**[♯], **Ting Liu**[♯], **William Yang Wang**[♭*]
[♯]Harbin Institute of Technology, Harbin, China
[♭]University of California, Santa Barbara, USA
{qfzhu, wnzhang, tliu}@ir.hit.edu.cn
{william}@cs.ucsb.edu

## Abstract

Learning a neural response generation model on data synthesized under the adversarial training framework helps to explore more possible responses. However, most of the data synthesized de novo are of low quality due to the vast size of the response space. In this paper, we propose a counterfactual off-policy method to learn on a better synthesis of data. It takes advantage of a real response to infer an alternative that was not taken using a structural casual model. Learning on the counterfactual responses helps to explore the high-reward area of the response space. An empirical study on the DailyDialog dataset shows that our approach significantly outperforms the HRED model as well as the conventional adversarial training approaches.

## 1 Introduction

Data-driven generation-based dialog system (Shang et al., 2015a; Vinyals and Le, 2015; Sordoni et al., 2015a) responds to users by learning from conversational data. Nevertheless, it suffers from data insufficient problem as there may exist many potential responses for a given message (Li et al., 2016a). Adversarial and reinforcement learning could alleviate the issue by training on trajectories (responses in our task) synthesized by the model itself (Li et al., 2017a, 2016b). However, a mismatch between the synthesis of data and the real conversational data makes the model hard to generalize to the real environment (Jiang and Li, 2016; Buesing et al., 2019).

In contrast, humans could respond better with limited responses in past experiences by counterfactual inference (Pearl, 2009). For instance, having observed a real response of a given message, one may naturally reason what would have been if he produces an alternate response, while everything else remains unchanged. Learning from the counterfactual alternate response reduces the mismatch between the synthesized and the real data as it is inferred in the environment where the real response occurs.

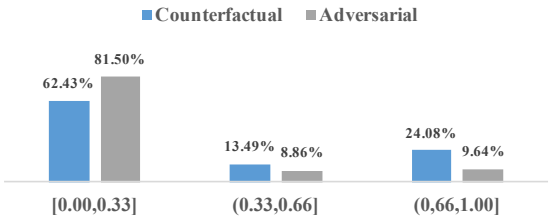


Figure 1: The distribution of trajectory reward of the standard adversarial learning approach and our counterfactual off-policy training method. The x-axis corresponds to three reward intervals. The y-axis is the percentage.

Motivated by this, we propose a counterfactual off-policy training (COPT) method for adversarial and reinforcement learning neural response generation. First, it makes use of an observed response from the real environment to infer its *scenario*, which captures all model-irrelevant aspects and is represented by random noise variables in a structural causal model (SCM) (Pearl, 2009). Then the COPT predicts an alternative of the observed response in the inferred scenario using the *casual mechanism*, a deterministic function in the SCM. After that, the COPT regards the alternate response as a counterfactual trajectory and learns on it.

Intuitively, the counterfactual trajectory is synthesized by grounding the model in the scenario inferred from the real response, rather than the scenario sampled from scratch in standard adversarial and reinforcement learning approaches. This improves the quality of the trajectory, as shown in Figure 1, and subsequently benefits response generation models. To verify the effectiveness of our

*Corresponding author.

approach, we conduct experiments on the public available DailyDialog dataset (Li et al., 2017b). Experimental results show that our approach significantly outperforms previous adversarial approaches in both automatic and human evaluations.

The contributions of this paper are summarized as follows:

- We introduce the counterfactual inference into the response generation by casting the model as a structural casual model.
- Our counterfactual trajectory is of higher quality than that synthesized from scratch in standard adversarial learning approaches.
- Experimental results show that our approach significantly outperforms previous adversarial training approaches in both automatic and human evaluations.

## 2 Related Work

**Response Generation** Data-driven dialogue systems can be roughly divided into two categories: retrieval-based (Leuski et al., 2006; Ji et al., 2014; Yan et al., 2016) and generation based (Shang et al., 2015b; Sordoni et al., 2015b; Vinyals and Le, 2015). Responses of retrieval-based methods come from a fixed candidate response set and thus are incapable of being customized. The generation-based methods can create new responses but the vanilla sequence to sequence models tends to produce generic responses.

One way to address the generic response problem is by introducing external knowledge, such as keywords (Mou et al., 2016), topics (Xing et al., 2017), and retrieved candidate responses (Song et al., 2018; Wu et al., 2019). Another way is to optimize the architecture of networks. There are two architectures widely employed in this research line: the variational auto-encoder (Bowman et al., 2016; Zhao et al., 2017) and the generative adversarial network (Goodfellow et al., 2014; Li et al., 2017a; Zhang et al., 2018; Xu et al., 2018; Tuan and Lee, 2019). Our approach falls into the latter category. The differences between our approach and other adversarial training approaches are as follows. First, we cast the response generation model as a structural casual model during the training process. Second, we learn on counterfactual trajectories that inference from the structural casual model given the real responses. Third, a pre-trained behavior policy is involved during the generation of our trajectories. This makes our approach an off-policy algorithm and benefits the exploration of possible responses.

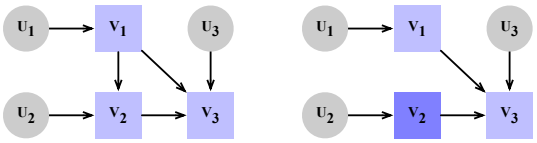


Figure 2: An example of an SCM. Left: An SCM with scenarios $\boldsymbol{U}$ and random variables $\boldsymbol{V}$. Each random variable $\boldsymbol{V}_i$ is determined by its parents, scenario $\boldsymbol{U}_i$, and a determinstic function $f_i$ (purple squares). Right: An intervention on the left SCM. The casual mechanism $\boldsymbol{V}_2 = f_2(\boldsymbol{V}_1, \boldsymbol{U}_2)$ is replaced by $\boldsymbol{V}_2 = f_2^I(\boldsymbol{U}_2)$.

**Counterfactual Inference** Counterfactual inference is a concept derived from psychology. It describes the human capacity to learn from past experience by reasoning alternate outcomes that could have been (Pearl and Mackenzie, 2018). Buesing et al. (2019) connect the concept with reinforcement learning to learn from off-policy data. Oberst and Sontag (2019) introduce a gumbel-max structural casual model to address non-identifiability issue. Kaushik et al. (2020) propose a data-augmentation method that edits the data with counterfactual labels, resulting in classifiers that are less sensitive to spurious patterns.

## 3 Method

We cast a response generation model as a structural causal model during the training process to learn on counterfactual trajectories. We will first review the concept of the structural causal model and then introduce our counterfactual off-policy training method.

### 3.1 Background: Structural Causal Model

A structural casual model (SCM) is a directed acyclic graph (DAG) over random variables $\boldsymbol{V} = \{\boldsymbol{V}_1, ..., \boldsymbol{V}_N\}$, independent noise random variables $\boldsymbol{U} = \{\boldsymbol{U}_1, ..., \boldsymbol{U}_N\}$ with distribution $P_U$, and deterministic functions $\boldsymbol{F} = \{f_1, ..., f_N\}$ such that $\boldsymbol{V}_i = f_i(\boldsymbol{PA}_i, \boldsymbol{U}_i)$, where $\boldsymbol{PA}_i$ are the parents of $\boldsymbol{V}_i$ in the DAG. $\boldsymbol{U}$ and $\boldsymbol{F}$ are also referred to scenarios and casual mechanisms, respectively. Figure 2 (Left) shows an example of an SCM. Each random variable $\boldsymbol{V}_i$ is generated by its parents, scenario $\boldsymbol{U}_i$, and a deterministic function $f_i$, e.g., $\boldsymbol{V}_2 = f_2(\boldsymbol{V}_1, \boldsymbol{U}_2)$.

We represent a response generation model by an SCM during the training process. Given an input message $\boldsymbol{X}$, we express the conditional probabilistic distribution of a response $P(\boldsymbol{Y}|\boldsymbol{X})$ as a deterministic function with independent noise random variable $\boldsymbol{U}$, such that $\boldsymbol{Y} = f_\pi(\boldsymbol{X}, \boldsymbol{U})$. We denote the causal mechanism as $f_\pi$ to highlight the role of the policy (parameters) of the model. The scenario $\boldsymbol{U}$ captures all unobserved model-free properties, like the user profile. The response generation SCM makes it possible to sample counterfactual responses (trajectories) under the same scenario with the real responses. Intuitively, the counterfactual response is produced by referring to the real response, rather than by sampling from scratch. This helps the model to explore the area of higher reward in the response space.

**Intervention in SCM** Given an SCM, an intervention $I$ consists of replacing some of the original $f_i(\boldsymbol{PA}_i, \boldsymbol{U}_i)$ with other functions $f_i^I(\boldsymbol{PA}_i^I, \boldsymbol{U}_i)$, where $\boldsymbol{PA}_i^I$ are the parents of $\boldsymbol{V}_i$ in a new DAG. Figure 2 (Right) shows an example of an intervention, where the original casual mechanism $\boldsymbol{V}_2 = f_2(\boldsymbol{V}_1, \boldsymbol{U}_2)$ is replaced with $\boldsymbol{V}_2 = f_2^I(\boldsymbol{U}_2)$.

Accordingly, intervention in our response generation SCM corresponds to the update of the policy. For example, the update from a behavior policy $\mu$ to the current policy $\pi$ is the intervention of replacing $\boldsymbol{Y} = f_\mu(\boldsymbol{X}, \boldsymbol{U})$ with $\boldsymbol{Y} = f_\pi(\boldsymbol{X}, \boldsymbol{U})$ (here, the intervention is conducted on the casual mechanism only, the parents of $\boldsymbol{Y}$ in the new DAG remain unchanged).

**Counterfactual Inference in SCM** Given an SCM and observed some variables $\boldsymbol{V}_o = \boldsymbol{v}_o$, counterfactual inference answers the following question: what the variables $\boldsymbol{V}_q$ would have been had we done the intervention $I$ while remaining everything else unchanged. Recall that we aim to sample a response under the scenario of a real response. It can be seen as querying what the response $\boldsymbol{Y}$ would have been had we follow the current policy $\pi$ rather than the policy $\mu$ that generates the real response.

Typically, the counterfactual inference answers the question in the following steps:

- estimate the posterior distribution of scenarios given real responses $P(\boldsymbol{U}|\boldsymbol{Y})$.
- sample $\boldsymbol{u}$ from $P(\boldsymbol{U}|\boldsymbol{Y})$.
- do interventions by switching the policy from $\mu$ to $\pi$.
- inference the counterfactual response by: $\hat{\boldsymbol{y}} = f_\pi(\boldsymbol{x}, \boldsymbol{u})$.

We will introduce the counterfactual inference in our model in more detail in the following section.

### 3.2 Counterfactual Off-Policy Training

Our method is built upon the adversarial training framework. It consists of a generator $G$ and a discriminator $D$.

**Generator** The generator $G$ is a sequence to sequence (Seq2Seq) model with the attention mechanism (Sutskever et al., 2014; Bahdanau et al., 2015). Given an input message, $G$ reads it into hidden states via an encoder LSTM (Hochreiter and Schmidhuber, 1997):

$$\boldsymbol{h_i} = \text{LSTM}(x_i, \boldsymbol{h_{i-1}}), \tag{1}$$

where $x_i$ is the $i$-th word of the message and $\boldsymbol{h_i}$ is the corresponding hidden state.

At the $j$-th decoding time step, $G$ first summarizes the hidden states of the encoder into a context vector using the attention mechanism:

$$\alpha_{ij} = \frac{\exp(q(\boldsymbol{s}_j, \boldsymbol{h}_i))}{\sum_{l=1}^{L} \exp(q(\boldsymbol{s}_j, \boldsymbol{h}_l))}, \tag{2}$$

$$\boldsymbol{c}_j = \sum_{i=1}^{L} \alpha_{ij} \boldsymbol{h}_i, \tag{3}$$

where $\boldsymbol{s}_j$ is the $j$-th hidden state of the decoder. $L$ is the length of the message, and $q$ is a feed-forward network. After that, the decoder predicts a distribution $\boldsymbol{p}_j$ over the vocabulary as follows:

$$\boldsymbol{s}_j = \text{LSTM}([\boldsymbol{e}(\hat{y}_{j-1}) : \boldsymbol{c_j}], \boldsymbol{s_{j-1}}), \tag{4}$$

$$\boldsymbol{p}_j = \text{softmax}(\boldsymbol{s}_j \cdot \boldsymbol{O}), \tag{5}$$

where $\boldsymbol{e}(\cdot)$ denotes the embedding of a word. $\hat{y}_{j-1}$ is the word generated in the previous time step. $\boldsymbol{O}$ is the output matrix.

In the standard generator of the adversarial training, each response word is sampled from $\boldsymbol{p}_j$ by: $\hat{Y}_j \sim P_\pi(\hat{Y}_j|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:j-1}) = \boldsymbol{p}_j$. To covert the probabilistic generation model $P_\pi(\hat{Y}_j|X, \hat{\boldsymbol{Y}}_{1:j-1})$ into our response generation SCM, we utilize the Gumbel-Max Trick (Oberst and Sontag, 2019) as follows:

$$\begin{aligned} \hat{Y}_j &= f_\pi(\boldsymbol{X}, \boldsymbol{Y}_{1:j-1}, \boldsymbol{U}_j) \\ &= \arg\max(\log P_\pi(Y_j|\boldsymbol{X}, \boldsymbol{Y}_{1:j-1}) + \boldsymbol{U}_j), \end{aligned} \tag{6}$$

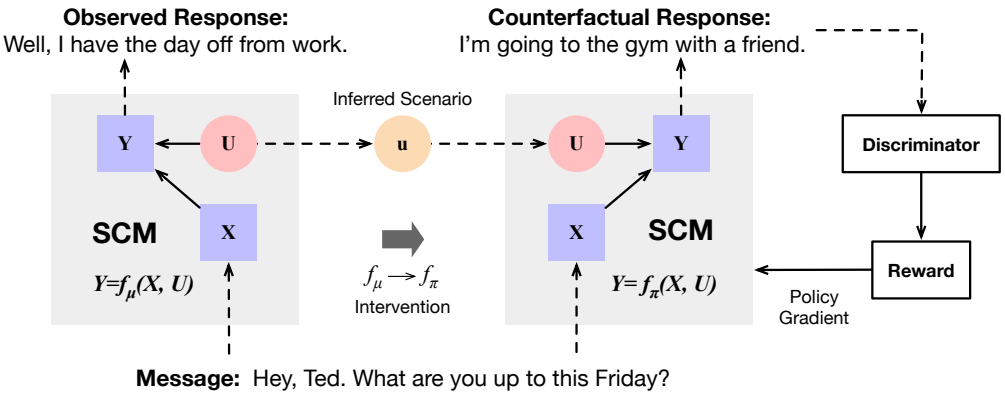


Figure 3: An example of our counterfactual off-policy training method. $\boldsymbol{\pi}$ is the current policy we aim to learn. $\boldsymbol{\mu}$ is the behavior policy that generates the observed real responses. First, we infer the scenario $\boldsymbol{u}$ where the observed real response occurs. Then we replace the casual mechanism in the SCM by switching the policy from $\boldsymbol{\mu}$ to $\boldsymbol{\pi}$. After that, the counterfactual response is reasoned by the intervened SCM in the inferred scenario.

where $\boldsymbol{U}_j$ follows the standard Gumbel distribution: $\boldsymbol{U}_j \sim \text{Gumbel}(0, 1)$.

Our method differs from the standard adversarial training in that it learns from counterfactual trajectories. We assume a real response derives from a behavior policy $\mu$ under the scenario $\boldsymbol{U} = \{\boldsymbol{U_1}, ..., \boldsymbol{U_L}\}$. The current policy $\pi$ is an intervention of $\mu$ and generates the counterfactual trajectory under the same scenario. Figure 3 shows an example of a step time during the generation process of the counterfactual trajectory. Taking the $j$-th time step as an example, we first estimate the posterior distribution of the scenario $P(\boldsymbol{U_j}|\boldsymbol{Y_j})$ and sample $\boldsymbol{u_i}$ from it. Intuitively, $\boldsymbol{u_j} \in \mathbb{R}^{|V|}$ is an vector where $\arg\max(\log \boldsymbol{p_j} + \boldsymbol{u_j}) = y_j$. Following Oberst and Sontag (2019), we assume $P(\boldsymbol{U_j}|\boldsymbol{Y_j})$ follows the Gumbel distribution and draw $\boldsymbol{u_j}$ using the rejection sampling. Concretely, we sample $\boldsymbol{u_j}$ from prior $P(\boldsymbol{U_j})$ (Gumbel(0, 1)) and reject those where $\arg\max(\log \boldsymbol{p_j} + \boldsymbol{u_j}) \neq y_j$. After that, we inference $\hat{y}_j$ by feeding $\boldsymbol{u}_j$ into the response generation SCM (Equation 6).

**Discriminator** The discriminator $D$ is introduced to provide a reward for each generation step. It takes as input a message $\boldsymbol{X}$, the word $\hat{\boldsymbol{Y}}_k$ produced in the generation step, and the partially generated response in the previous steps. The output reward $D(\hat{Y}_k|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:k-1})$ is a scalar ranging from 0 to 1. Concretely, $D$ first reads $\boldsymbol{X}$ and $\hat{\boldsymbol{Y}}_{1:k}$ with an encoder-decoder model. Then the last hidden state of the decoder is sent to a multi-Layer perceptron (MLP) to compute the reward.

**Adversarial Training** $G$ and $D$ are adversarially trained, where $G$ tries to fool $D$ by generating human-like responses while $D$ aims at distinguishing between machine-generated and real responses. Since a response is a sequence of discrete tokens, we pass by the gradient of $D$ to $G$ using the policy gradient algorithm. In this way, the generator is regarded as an agent, and its parameters define a policy. At each generation step, it takes an action by producing a word based on its state, which is defined as the partial response generated so far. Then it observes a reward from $D$ and updates its policy accordingly.

The goal of the generator is to minimize the negative expected reward: $J_G(\theta) = -\mathbb{E}_{\hat{\boldsymbol{Y}}_{1:k} \sim G} D(\hat{Y}_k|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:k-1})$, where $\theta$ is the parameters of $G$. Using the likelihood ratio trick (Williams, 1992), the gradient of $\theta$ can be derived as:

$$\begin{aligned}\bigtriangledown J_G(\theta) = &- \mathbb{E}_{\hat{Y}_k \sim G} D(\hat{Y}_k|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:k-1}) \\ &\cdot \bigtriangledown \log G_\theta(\hat{Y}_k|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:k-1}),\end{aligned} \quad (7)$$

where $G_\theta(\hat{Y}_k|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:k-1})$ is the probability of generating $\hat{Y}_k$ given $\boldsymbol{X}$ and $\hat{\boldsymbol{Y}}_{1:k-1}$.

The discriminator distinguishes between machine-generated and real responses. It aims to minimize the classification error rate by the following loss function:

$$\begin{aligned}J_D(\phi) = &- \mathbb{E} \log D(Y_k|\boldsymbol{X}, \boldsymbol{Y}_{1:k-1}) \\ &- \mathbb{E} \log(1 - D(\hat{Y}_k|\boldsymbol{X}, \hat{\boldsymbol{Y}}_{1:k-1})).\end{aligned} \quad (8)$$

Note that both $G$ and $D$ are pre-trained before adversarial training. First, $G$ is pre-trained on the training set with MLE loss. Then we sample a

**Algorithm 1** Counterfactual Off-Policy Training

**Require:**
The training set $\{\boldsymbol{X}, \boldsymbol{Y}\}$;
**Ensure:**
$\boldsymbol{\theta}_\pi$, parameters of the update policy $\pi$;
$\boldsymbol{\theta}_\mu$, parameters of the behavior policy $\mu$;
$\boldsymbol{\phi}$, parameters of the discriminator;

Randomly initialize $\boldsymbol{\theta}_\pi$, $\boldsymbol{\theta}_\mu$, and $\boldsymbol{\phi}$;
Pre-train $\pi$ and $\mu$ with MLE loss;
Generate responses using the pre-trained $\pi$;
Pre-train $D$ using machine-generated responses as negative samples and real responses as positive samples;
**for** epoch in number of epochs **do**
  **for** $g$ in g-steps **do**
    compute $P(\boldsymbol{Y}|\boldsymbol{X})$ under the policy$\mu$;
    sample a scenario $\boldsymbol{u}$ from $P(\boldsymbol{U}|\boldsymbol{Y})$;
    generate a counterfactual trajectory $\hat{\boldsymbol{Y}}$ with the scenario $\boldsymbol{u}$;
    optimize $\boldsymbol{\theta}_\pi$ on the pair $(\boldsymbol{X}, \hat{\boldsymbol{Y}})$;
  **end for**
  **for** $d$ in d-steps **do**
    Sample $\hat{\boldsymbol{Y}}$ with $\pi$ as a negative sample;
    Sample $\boldsymbol{Y}$ from the real responses as a positive sample;
    Update $\phi$ according to Equation 8;
  **end for**
**end for**
**return** $\theta, \phi$;

machine-generated response by the pre-trained $G$ for each message in the training set. After that, $D$ is pre-trained using the real responses as positive samples and the machine-generated responses as negative samples. The overall algorithm of the counterfactual off-policy training is summarized as Algorithm 1.

## 4 Experiments

### 4.1 Data

The experiments are conducted on the DailyDialog dataset. It is a multi-turn dataset and covers various topics of our daily life. Collected from websites for English learners, all its dialogues are human-written and usually end after reasonable turns. The dataset has already been divided into training, validation, and test set as shown in Table 1. For each $K$-turn dialogue, we split it into $K$-1 instances. Each instance contains at most three continuous utterances, where the last utterance is the response and the other utterances are concatenated as the message.

| | |
|---|---|
| Training Dialogues | 11,118 |
| Validation Dialogues | 1,000 |
| Test Dialogues | 1,000 |
| Average Speaker Turns Per Dialogue | 7.9 |
| Average Tokens Per Dialogue | 114.7 |
| Average Tokens Per Utterance | 14.6 |

Table 1: Statistics of the DailyDialog dataset.

### 4.2 Training Details

Our approach is implemented under the OpenNMT (Klein et al., 2017), an open-source framework for building sequence to sequence models. The vocabulary consists of the most frequent 10,000[1] words. Word embeddings are pre-trained using Glove (Pennington et al., 2014). Both of the encoder and the decoder are a 2-layer LSTM, whose number of hidden units is 500. The batch size is set to 64.

Following previous work (Li et al., 2017a), we pre-train the generator's policy $\pi$ using the MLE loss. Meanwhile, the behavior policy $\mu$ is pre-trained in the same way[2] as it aims at generating the real responses in the training set. During the adversarial training process, we use ADAM optimizer and initialize the learning rate to 1e-5. $G$ and $D$ are alternately trained for 1 batch and 5 batches.

### 4.3 Baselines

To verify the effectiveness of our method, the following baselines are compared:

- HRED (Serban et al., 2016): The hierarchical recurrent encoder-decoder.
- REGS (Li et al., 2017a): Reward for every generation step. The discriminator is trained on partially generated responses to provide a reward for each generation step.
- DPGAN (Xu et al., 2018): The diversity-promoting GAN introduces a language model based discriminator. The reward for every generation step is cross-entropy.

[1] Using a bigger vocabulary table (even the complete 17,438 words) observes no improvements on the validation loss but takes more time for training.

[2] $\pi$ and $\mu$ are pre-trained independently with different initial parameters.

| Model | Dist-1 | # of UNI | Dist-2 | # of BI | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|---|---|---|---|---|---|---|---|---|
| HRED (Serban et al., 2016) | 0.011 | 918 | 0.045 | 3,875 | 33.0 | 4.5 | 1.1 | 0.3 |
| REGS (Li et al., 2017a) | 0.021 | 1,205 | 0.097 | 5,552 | 38.4 | 6.8 | 2.0 | 0.7 |
| DPGAN (Xu et al., 2018) | 0.002 | 225 | 0.008 | 1,034 | 31.6 | 3.7 | 0.4 | 0.1 |
| StepGAN (Tuan and Lee, 2019) | 0.013 | 1,063 | 0.065 | 5,283 | 36.1 | 6.6 | 1.9 | 0.6 |
| Ours | 0.026 | 1,398 | 0.116 | 6,234 | 39.8 | 7.7 | 2.3 | 0.8 |

Table 2: Automatic evaluation results of the number of distinct uni-grams (# of UNI) and bi-grams (# of BI), distinct-1 (Dist-1), distinct-2 (Dist-2), and BLEU scores.

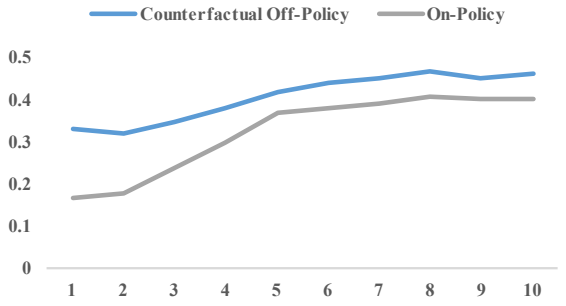


Figure 4: The average of trajectory reward of our counterfactual off-policy and the on-policy method. The x-axis corresponds to the epochs. The y-axis is the average.

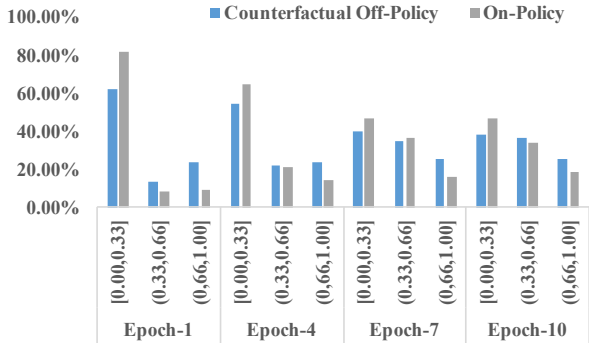


Figure 5: Trajectory reward distribution of our counterfactual off-policy and the on-policy method. The x-axis corresponds to three reward intervals. The y-axis is the percentage.

- StepGAN (Tuan and Lee, 2019): The stepwise GAN optimizes the discriminator by maximizing the average of state-action values of real responses. During the adversarial training process, the discriminator assigns scores for every generation step in the same way as REGS.

Our method differs from previous adversarial training approaches in that we represent the response model as an SCM and train it on the counterfactual trajectories.

### 4.4 Evaluation Metrics

#### Automatic Evaluation

We evaluate the diversity of our method using the *Distinct* metric (Li et al., 2016a) and the *BLEU* metric. Concretely, the Distinct-$k$ is the number of distinct $k$-grams normalized by the number of words of responses.

#### Human Evaluation

The human evaluation is conducted on 200 instances randomly sampled from the test set. Three annotators are employed to select a better response between our method and each baseline. Each annotator evaluates all the instances independently. Their consistency is measured by Fleiss Kappa (Fleiss, 1971).

### 4.5 Results

Table 2 shows the results of automatic evaluation. Our approach, REGS, and StepGAN significantly outperform HRED in Dist-1 and Dist-2. This indicates that adversarial training helps improve the diversity, which is in line with the observation in previous work (Li et al., 2017a). There is no improvement in DPGAN compared with HRED in our experiments. We believe this is because the scale of the DailyDialog dataset is not large enough for sufficiently training the language model based discriminator in DPGAN. By contrast, our approach outperforms other adversarial training approaches in both Dist-1 and Dist-2, indicating that our counterfactual off-policy method is more effective in improving diversity. The results of BLEU scores are shown in Table 2. Our approach has the highest scores from BLEU-1 to BLEU-4, which demonstrates its effectiveness in generating relevant responses.

To further analyze the effectiveness of our counterfactual off-policy in exploring the high-reward area of the possible response space, we randomly sample 10,000 messages from the training set[3] and

[3]The messages are sampled from the training set, rather than the test set or validation set, because we aim to analyze

| | |
|---|---|
| Message | How much is it ? <EOS> It's free of charge. |
| HRED | 25 cents. |
| REGS | Here you go. |
| DPGAN | Know. Me ,I am not sure. |
| StepGAN | Here is 200 yuan. |
| Ours | Does it include utilities? |
| Message | Perhaps you'd be interested in red roses. <EOS> What would the roses cost me ? |
| HRED | Of course. I 'm not sure. |
| REGS | Twenty dollars. |
| DPGAN | Think it's a very good idea. Me, I am sure he will be. |
| StepGAN | $20. |
| Ours | Twenty cents altogether. |

Table 3: Some examples of responses generated by baselines and our approach.

compare the reward of trajectories of various approaches. Figure 4 and 5 show the average and the distribution of trajectory rewards of our counterfactual off-policy and the on-policy method, respectively. Our approach has a higher average reward and more high-reward trajectories.

### 4.6 Case Study

Table 3 shows some examples of responses generated by the baselines and our approach. HRED tends to produce generic responses, like "Of course. I'm not sure" in the second example. The responses of DPGAN sometimes are not fluent. We believe this is because the scale of the DailyDialog dataset is not large enough for sufficiently training the language model discriminator. While responses of our approach are more informative and contain more specific content, like the discussion about the "utilities" in the first example.

## 5 Conclusion

We propose a counterfactual off-policy training method for neural response generation in dialogue systems. In contrast to existing approaches, our approach learns on counterfactual trajectories inferred from the structural casual model in the scenario where the real responses occur. This helps the model to explore the high-reward area of the possible response space. Experiments show that the counterfactual off-policy method significantly improves the quality of the generated responses, which demonstrates the effectiveness of this approach.

the trajectory that a model learns during the training process.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Lars Buesing, Theophane Weber, Yori Zwols, Nicolas Heess, Sebastien Racaniere, Arthur Guez, and Jean-Baptiste Lespiau. 2019. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *ICLR*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Journal of Psychological bulletin*, 76(5):378.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the Twenty-eighth Conference on Neural Information Processing Systems*, pages 2672–2680.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Journal of Neural computation*, 9(8):1735–1780.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Nan Jiang and Lihong Li. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *Proceedings of ICLR 2017)*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. 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, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3349–3358.

Michael Oberst and David Sontag. 2019. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic Books.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015a. Neural responding machine for short-text conversation. 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 1577–1586, Beijing, China. Association for Computational Linguistics.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015b. Neural responding machine for short-text conversation. 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)*, pages 1577–1586.

Yiping Song, Rui Yan, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, and Dongyan Zhao. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015a. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015b. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Twenty-Eighth Conference on Neural Information Processing Systems*, pages 3104–3112.

Yi-Lin Tuan and Hung-Yi Lee. 2019. Improving conditional sequence generative adversarial networks by stepwise evaluation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(4):788–798.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

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

Yu Wu, Furu Wei, Shaohan Huang, Zhoujun Li, and Ming Zhou. 2019. Response generation by context-aware prototype editing. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, volume 17, pages 3351–3357.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the Thirty-Second Conference on Neural Information Processing Systems*, pages 1810–1820.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.