De-Confounded Variational Encoder-Decoder for Logical Table-to-Text Generation

Wenqing Chen[1,2], **Jidong Tian**[1,2], **Yitian Li**[1,2], **Hao He**[1,2*], **Yaohui Jin**[1,2*]
[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]State Key Lab of Advanced Optical Communication System and Network,
Shanghai Jiao Tong University
{wenqingchen, frank92, yitian_li, hehao, jinyh}@sjtu.edu.cn

Abstract

Logical table-to-text generation aims to automatically generate fluent and logically faithful text from tables. The task remains challenging where deep learning models often generated linguistically fluent but logically inconsistent text. The underlying reason may be that deep learning models often capture surface-level spurious correlations rather than the causal relationships between the table $\boldsymbol{x}$ and the sentence $\boldsymbol{y}$. Specifically, in the training stage, a model can get a low empirical loss without understanding $\boldsymbol{x}$ and use spurious statistical cues instead. In this paper, we propose a de-confounded variational encoder-decoder (DCVED) based on causal intervention, learning the objective $p(\boldsymbol{y}|\text{do}(\boldsymbol{x}))$. Firstly, we propose to use variational inference to estimate the confounders in the latent space and cooperate with the causal intervention based on Pearl's do-calculus to alleviate the spurious correlations. Secondly, to make the latent confounder meaningful, we propose a back-prediction process to predict the not-used entities but linguistically similar to the exactly selected ones. Finally, since our variational model can generate multiple candidates, we train a table-text selector to find out the best candidate sentence for the given table. An extensive set of experiments show that our model outperforms the baselines and achieves new state-of-the-art performance on two logical table-to-text datasets in terms of logical fidelity.

1 Introduction

Data-to-text generation refers to the task of generating descriptive text from non-linguistic inputs. With the different types of inputs, this task can be defined more specifically, such as abstract meaning representation to text (Zhao et al., 2020; Bai et al., 2020a), infobox with key-value pairs to text (Bai et al., 2020b), graph-to-text (Song et al., 2020), and table-to-text (Wang et al., 2020; Parikh et al., 2020) generation.

Among these tasks, we focus on logical table-to-text generation, which aims to generate fluent and logically faithful text from tables (Chen et al., 2020a). And the ability of logical inference is a kind of high-level intelligence, which is non-trivial for text generation systems in reality. The task remains challenging because the reference sentences often convey logically inferred information, which is not explicitly presented in the table. As a consequence, data-driven models often generated linguistically fluent but logically inconsistent text. Recent progress on this task mainly lies in the use of pretrained language models (LMs) like GPT-2 (Radford et al., 2018), which was shown to perform much better than non-pretrained models (Chen et al., 2020a,e).

However, it is still arguable that whether pretrained LMs can correctly capture the logics, as pretrained LMs like BERT would use spurious statistical cues for inference (Niven and Kao, 2019). The substantial difficulty for this task does not lay on whether to use the pretrained models or not. Instead, the difficulty is because the surface-level spurious correlations are easier to capture than the causal relationship between the table and the text. For example, we have observed that a model cooperating with GPT-2 generated a sentence "*The album was released in the United States 2 time*" for a given table. But the country where the album was released twice is "*the United Kingdom*"[1]. In the training stage, a model may get low training loss by utilizing the surface-level correlations without

* Corresponding Authors

[1]The details of the table can be found in Section 5.6

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5532–5542
August 1–6, 2021.

actually focusing on the selected entities. As a result, in the inference stage, the model is possible to produce incorrect facts.

In this paper, we view the logical table-to-text generation from the perspective of causal inference and propose a de-confounded variational encoder-decoder (DCVED). Firstly, given the table-sentence pair $(\boldsymbol{x}, \boldsymbol{y})$, we assume confounders $\boldsymbol{z}_c$ existed in the latent space and contributing to the surface-level correlations (e.g., "*the United States*" and "*the United Kingdom*"). We estimate $\boldsymbol{z}_c$ in the latent space based on variational inference, and cooperate the causal intervention based on Pearl's do-calculus (Pearl, 2010) to learn the objective $p(\boldsymbol{y}|\text{do}(\boldsymbol{x}))$ instead of $p(\boldsymbol{y}|\boldsymbol{x})$. Secondly, to make the latent confounder meaningful, we propose a back-prediction process to ensure the latent confounder $\boldsymbol{z}_c$ can predict the not-used entities but linguistically similar to the exactly selected ones. We also consider the exactly selected entities as the mediators in our de-confounded architecture models. Finally, since our variational model can generate multiple candidates, we train a table-text selector to find out the best text for the table. An extensive set of experiments show that our model achieves new state-of-the-art performance on two logical table-to-text datasets in terms of logical fidelity.

The main contributions of this work can be summarized as follows:

- We propose to use variational inference to estimate the confounders in the latent space and cooperated with back-prediction to make the latent variable meaningful.
- We propose a generate-then-select paradigm jointly considering the surface-level and logical fidelity, which can be considered as an alternative to reinforcement learning.
- The experiments have shown that our model achieves new state-of-the-art performance on two logical table-to-text datasets with or without pretrained LMs.

2 Related Work

Table-to-Text Generation. The task of table-to-text generation belongs to the data-to-text generation, where a key feature is the structured input data. Lebret et al. (2016) used a seq2seq neural model with a field-infusing strategy that obtains field-position-aware and field-words-aware cell embeddings to generate sentences from Wikipedia tables. A follow-up work proposed to update the cell memory of the LSTM by a field gate to help LSTM identify the boundary between different cells (Liu et al., 2018). Transformer-based (Vaswani et al., 2017) models were also proposed which improved the ability to capture long-term dependencies between cells (Ma et al., 2019; Wang et al., 2020; Chen et al., 2020a). It is worth to mention that the copy mechanism (Luong et al., 2015) is an important part to deal with the out-of-vocabulary (OOV) words (Lebret et al., 2016; Gehrmann et al., 2018; Chen et al., 2020a) when not using pretrained language models.

Logical Table-to-Text Generation. While usually fluent, existing methods often hallucinate phrases that contradict the facts in the table. To benchmark models' ability to generate logically consistent sentences, recent work proposed a dataset collected from open domain (Chen et al., 2020a), which would score low on those models ignoring logical consistency. Follow-up work further proposed another dataset that involved logical forms as additional supervision information (Chen et al., 2020e), which includes common logic types paired with the underlying logical forms.

Causal Inference. Machine learning models often suffer from the spurious statistical correlations brought by unmeasured or latent confounders (Keith et al., 2020). To eliminate the confounding bias, one approach is applying the causal intervention based on Pearl's do-calculus (Pearl, 2010). However, it remains an open problem to choose proper confounders, and the language of text itself could be a confounder (Keith et al., 2020). It is worth noting that high-quality observations of the mediators can also reduce the confounding bias, as the models will reduce the possibility of counting on the confounders (Chen et al., 2020d).

3 Backgrounds

Before introducing our models, we briefly review the framework of VAE (Kingma and Welling, 2014), a generative model which allows to generate high-dimensional samples from a continuous space. In the probability model framework, the probability of data $\boldsymbol{x}$ can be computed by:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \quad (1)$$

where it is approximated by maximizing the evidence lower bound (ELBO):

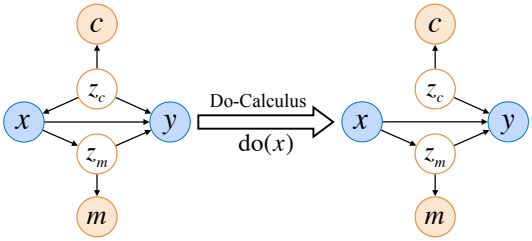
$$\log p_\theta(\boldsymbol{x}) \geq \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \quad (2)$$

where $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ denotes the decoder with parameters θ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is obtained by an encoder with parameters ϕ, and $p(\boldsymbol{z})$ is a prior distribution, for example, a Gaussian distribution. And $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) Divergence between two distributions.

When applied to seq2seq generation where the input and the output are denoted by $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, the conditional variational auto-encoder (CVAE), or often known as variational encoder-decoder (VED), is used with following approximation:

$$\log p_\theta(\boldsymbol{y}|\boldsymbol{x}) \geq \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})\|p(\boldsymbol{z}|\boldsymbol{x})) \quad (3)$$

In the vanilla CVAE formulation, such as the ones adopted in (Kingma et al., 2014; Jain et al., 2017), the prior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ is approximated to $p(\boldsymbol{z})$, which is independent on $\boldsymbol{x}$ and fixed to a zero-mean unit-variance Gaussian distribution $\mathcal{N}(0, \mathrm{I})$. However, this formulation is shown to induce a strong model bias (Tomczak and Welling, 2018) and empirically perform worse than non-variational models (Wang et al., 2017) in multi-modal situation.



(a) Mediation-Confounding (b) De-Confounded

Figure 1: The causal graphs before and after the *do-calculus*. The symbols $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}_m$, $\boldsymbol{z}_c$ denote the input table, the output sentence, the hidden mediator, and the hidden confounder, respectively. We assume $\boldsymbol{c}$ and $\boldsymbol{m}$ to be the proxy variables of $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$, respectively, which are relatively easy to be observed.

4 Methodology

4.1 De-Confounded VED

From a human perspective, multiple sentences can properly describe a given table, varying with different concerns, different logical types or linguistic realizations. Therefore, given the input data $\boldsymbol{x}$ and the output sentence $\boldsymbol{y}$, we can assume a latent variable $\boldsymbol{z}$ existed leading to a conditional generation process $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$ where $\boldsymbol{z}$ contributes to the diversity. It suggests a CVAE framework with Equation 3. However, as discussed in Section 3, the vanilla CVAE will introduce a model bias (Tomczak and Welling, 2018). In this subsection, we re-think the CVAE from the perspective of causal inference. We assume a directed acyclic graph (DAG) existed, which includes a mediator $\boldsymbol{z}_m$ and a confounder $\boldsymbol{z}_c$ as shown in Figure 1(a). The mediator is determined by $\boldsymbol{x}$ and has causal effects on $\boldsymbol{y}$, while the confounder has causal effects on both $\boldsymbol{x}$ and $\boldsymbol{y}$.

When only considering $\boldsymbol{z}_m$, we can compute the probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ by:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &= \sum_{\boldsymbol{z}_m} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_m)p(\boldsymbol{z}_m|\boldsymbol{x}) \\ &= \mathbb{E}_{\boldsymbol{z}_m\sim p_\varphi(\boldsymbol{z}_m|\boldsymbol{x})}p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_m) \end{aligned} \quad (4)$$

where φ denotes the parameters of a mediator predictor. An example for $\boldsymbol{z}_m$ is the selected entity (e.g., United Kingdom) from the table $\boldsymbol{x}$ and exactly appeared in $\boldsymbol{y}$. The vanilla CVAE will constrain $\boldsymbol{z}_m$ in the continuous space, and further approximate the prior distribution $p(\boldsymbol{z}_m|\boldsymbol{x})$ to $p(\boldsymbol{z}_m)$, which produces biased information.

However, it does not mean that removing the approximation between $p(\boldsymbol{z}_m|\boldsymbol{x})$ and $p(\boldsymbol{z}_m)$ is enough. We observe that models often rely on spurious statistical cues for prediction, resulting in some linguistically similar but inconsistent expressions in the generated sentences (e.g., using "*The United States*" instead of "*The United Kingdom*). The model is possible to minimize the training loss relying on the surface-level correlations between the selected entity and the high-frequency entity. In this case, the high-frequency entity belongs to the confounder $\boldsymbol{z}_c$. In the inference stage, model may infer contradicting facts due to a high posterior probability of $q(\boldsymbol{z}_c|\boldsymbol{x})$.

To eliminate the spurious correlations, we apply causal intervention by learning the objective $p(\boldsymbol{y}|\mathrm{do}(\boldsymbol{x}))$ instead of $p(\boldsymbol{y}|\boldsymbol{x})$, which forces the input to be the observed data $\boldsymbol{x}$, and removes all the arrows pointing to $\boldsymbol{x}$ as shown in Figure 1(b). When only considering $\boldsymbol{z}_c$, we can compute the in-

tervened probability distribution by:

$$
\begin{aligned}
p(\boldsymbol{y}|\text{do}(\boldsymbol{x})) &= \sum_{\boldsymbol{z}_c} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_c)p(\boldsymbol{z}_c) \\
&= \mathbb{E}_{\boldsymbol{z}_c \sim p(\boldsymbol{z}_c)} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_c)
\end{aligned} \tag{5}
$$

where $\boldsymbol{z}_c$ is no longer determined by $\boldsymbol{x}$, making $p(\boldsymbol{z}_c|\text{do}(\boldsymbol{x})) = p(\boldsymbol{z}_c)$. When applying variational inference to $\boldsymbol{z}_c$, we have:

$$
\begin{aligned}
p(\boldsymbol{y}|\text{do}(\boldsymbol{x})) \geq\ & \mathbb{E}_{\boldsymbol{z}_c \sim q_\phi(\boldsymbol{z}_c|\boldsymbol{y})} p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_c) \\
& - \text{KL}(q_\phi(\boldsymbol{z}_c|\boldsymbol{y})|p(\boldsymbol{z}_c))
\end{aligned} \tag{6}
$$

It can be seen that the confounder $\boldsymbol{z}_c$ is more suitable than the mediator $\boldsymbol{z}_m$ to cooperate with variational inference, as cutting off the link $\boldsymbol{z}_c \rightarrow \boldsymbol{x}$ will naturally make $p(\boldsymbol{z}_c|\text{do}(\boldsymbol{x}))$ to $p(\boldsymbol{z}_c)$.

When jointly considering $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$, we have:

$$
\begin{aligned}
&p(\boldsymbol{y}|\text{do}(\boldsymbol{x})) = \sum_{\boldsymbol{z}_m} \int_{\boldsymbol{z}_c} p(\boldsymbol{y}, \boldsymbol{z}_m, \boldsymbol{z}_c|\text{do}(\boldsymbol{x}))\text{d}\boldsymbol{z}_c \\
&\geq \mathbb{E}_{\boldsymbol{z}_m \sim p_\varphi(\boldsymbol{z}_m|\boldsymbol{x}), \boldsymbol{z}_c \sim q_\phi(\boldsymbol{z}_c|\boldsymbol{y})} [\log p_\theta(\boldsymbol{y}|\boldsymbol{x}, \\
&\boldsymbol{z}_m, \boldsymbol{z}_c)] - \text{KL}(q_\phi(\boldsymbol{z}_c|\boldsymbol{y})\|p(\boldsymbol{z}_c))
\end{aligned} \tag{7}
$$

according to the intervened causal graph in Figure 1(b). The symbols ϕ, φ and θ denote the parameters of three probability modeling networks, respectively. It is worth noting that we do not apply variational inference to $\boldsymbol{z}_m$ because finding a proper prior distribution $p(\boldsymbol{z}_m|\boldsymbol{x})$ remains another big topic. Instead, our framework is easy to implement.

4.2 Making Latent Variables Meaningful

However, there is no guarantee that $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$ can represent the real mediators and confounders in Equation 7. If we have no other observed variables, the confounder $\boldsymbol{z}_c$ would mainly represent the covariate which is naturally independent of $\boldsymbol{x}$ and has causal effects on $\boldsymbol{y}$.

Therefore, we further involve proxy variables $\boldsymbol{m}$ and $\boldsymbol{c}$ for $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$, respectively, where the full causal graph is shown in Figure 1. Proxy variables are the proxies of hidden or unmeasured variables (Miao et al., 2018). In practice, the mediators and the confounders are often too complex and can not be directly observed. For example, we may not be able to directly measure one's socioeconomic status but we are probable to get a proxy by the zip code or job type (Louizos et al., 2017). To make the latent variables $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$ meaningful, we add two additional networks and the learning objective is maximizing:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}_m \sim p_\varphi(\boldsymbol{z}_m|\boldsymbol{x}), \boldsymbol{z}_c \sim q_\phi(\boldsymbol{z}_c|\boldsymbol{y})} [\log p_\theta(\boldsymbol{y}|\boldsymbol{x}, \\
&\boldsymbol{z}_m, \boldsymbol{z}_c)] - \text{KL}(q_\phi(\boldsymbol{z}_c|\boldsymbol{y})\|p(\boldsymbol{z}_c)) \\
&+ \mathbb{E}_{\boldsymbol{z}_m \sim p_\varphi(\boldsymbol{z}_m|\boldsymbol{x})} [\log p_\Phi(\boldsymbol{m}|\boldsymbol{z}_m)] \\
&+ \mathbb{E}_{\boldsymbol{z}_c \sim q_\phi(\boldsymbol{z}_c|\boldsymbol{y})} [\log p_\Psi(\boldsymbol{c}|\boldsymbol{z}_c)]
\end{aligned} \tag{8}
$$

where Φ and Ψ denote the parameters of the two additional networks.

Back-Prediction from the Confounder. As shown in Figure 1(a), the confounder $\boldsymbol{z}_c$ inferred from $\boldsymbol{y}$ also have a causal effect on $\boldsymbol{x}$. Otherwise, the confounder will collapse into the covariate. The spurious correlations we have observed are that models often generate linguistically similar but logically inconsistent outputs. For example, "*the United Kingdom*" and "*the United State*" instead of "*the United Kingdom*" because the two entities are linguistically similar to each other. Therefore, we assume the proxy confounders $\boldsymbol{c}$ to be the not-mentioned entities in the given table. And we keep those high-frequency entities in the training set (≥ 5 times). Let $\boldsymbol{c} = \{c_{i,j}\} \in \mathbb{R}^{N_c \times L_c}$ where $c_{i,j}$ denotes the j-th token of i-th entity, and N_c and L_c denote the number of entities and maximum length of the entity, respectively. The log-probability $\log p_\Psi(\boldsymbol{c}|\boldsymbol{z}_c)$ is computed by:

$$
\log p_\Psi(\boldsymbol{c}|\boldsymbol{z}_c) = \sum_{i,j} \log p_\Psi(c_{i,j}|\boldsymbol{z}_c, \boldsymbol{c}_{i,<j}) \tag{9}
$$

where $\boldsymbol{c}_{i,<j}$ denotes the tokens preceding to the j-th token in the i-th entity. Then we minimize the cross-entropy between $p_\Psi(\boldsymbol{c}|\boldsymbol{z}_c)$ and $p(\boldsymbol{c})$.

Supervision for the Mediator. In the logical table-to-text generation task, from the human perspective, the correct mediators may be the selected entities, the logical types, or the logical forms (Chen et al., 2020e). In this paper, we only consider the selected entities as it is relatively easy to extract while the logical types or forms are labor-intensive to annotate. We represent the selected entities by $\boldsymbol{m} = \{m_{i,j}\} \in \mathbb{R}^{N_m \times L_m}$ where $m_{i,j}$ denotes the j-th token of i-th entity, and N_m and L_m denote the number of entities and maximum token number of the entity, respectively. The log-probability $p_\Phi(\boldsymbol{m}|\boldsymbol{z}_m)$ is computed by:

$$
\log p_\Phi(\boldsymbol{m}|\boldsymbol{z}_m) = \sum_{i,j} \log p_\Phi(m_{i,j}|\boldsymbol{z}_m, \boldsymbol{m}_{i,<j}) \tag{10}
$$

where $\boldsymbol{m}_{i,<j}$ denotes the tokens preceding to the j-th token in the i-th entity.

4.3 Encoders and Decoders Implementation

Then we introduce the implementations of $p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_m, \boldsymbol{z}_c)$, $p_\varphi(\boldsymbol{z}_m|\boldsymbol{x})$, and $q_\phi(\boldsymbol{z}_c|\boldsymbol{y})$. We assume that the seq2seq model consists of an encoder Enc$(\cdot)$ and a decoder Dec$(\cdot)$ for $p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_m, \boldsymbol{z}_c)$. And a target-oriented encoder T-Enc$(\cdot)$ is used for $q_\phi(\boldsymbol{z}_c|\boldsymbol{y})$.

Firstly, we need to implement $p_\varphi(\boldsymbol{z}_m|\boldsymbol{x})$ and $q_\phi(\boldsymbol{z}_c|\boldsymbol{y})$. Let $\boldsymbol{H}^x$ be the hidden states of $\boldsymbol{x}$ encoded via $\boldsymbol{H}^x = \text{Enc}(\boldsymbol{x})$, and $\boldsymbol{E}^y$ be the embeddings of $\boldsymbol{y}$ before fed to the decoder Dec$(\cdot)$. We use a fully-connected neural network (FCNN) to project $\boldsymbol{H}^x$ followed with the average pooling to obtain $\boldsymbol{z}_m$. And we use the target-oriented encoder to encode $\boldsymbol{E}^y$ and obtain $\boldsymbol{H}^y = \text{T-Enc}(\boldsymbol{E}^y)$. We apply the mean pooling operation to $\boldsymbol{H}^y$ and obtain $\boldsymbol{h}^y$. To modeling $q_\phi(\boldsymbol{z}_c|\boldsymbol{y})$ which is approximated to a Gaussian distribution, we use two FCNNs to process $\boldsymbol{h}^y$ and obtain the mean vector μ^y and the log variance $\log \sigma_y^2$ which makes:

$$q_\phi(\boldsymbol{z}_c|\boldsymbol{y}) = \mathcal{N}(\mu_y, \sigma_y^2) \tag{11}$$

To implement $p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_m, \boldsymbol{z}_c)$, our model cooperates an non-pretrained model "Field-Infusing+Trans" (Chen et al., 2020a) or a pretrained model "GPT-TabGen" (Chen et al., 2020a). Specifically, "Field-Infusing+Trans" uses an infusing field embedding network to produce header-words-aware and cell-position-aware embeddings $\boldsymbol{E}^p$, then concatenate $\boldsymbol{E}^p$ with token embeddings to obtain the infused embeddings $\boldsymbol{E} = \{\boldsymbol{e}_i\} \in \mathbb{R}^{L_t \times d}$ where $\boldsymbol{e}_i$ denotes the embedding of i-th token in the table $\boldsymbol{x}$, and L_t and d denote the token number and the dimension, respectively. Then the decoder is used to decode $\boldsymbol{y}$ token by token: $\boldsymbol{y}_t = \text{Dec}(\boldsymbol{H}^x, \boldsymbol{y}_{\le t}, \boldsymbol{z}_m, \boldsymbol{z}_c)$. The latent variables $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$ are concatenated as one latent variable and projected by a FCNN to get a vector $\boldsymbol{z}_{m,c}$ which has the same dimension with $\boldsymbol{H}^x$. Then we add $\boldsymbol{z}_{m,c}$ with $\boldsymbol{E}^y$ at each decoding step. When cooperated with "GPT-TabGen", the difference from "Field-Infusing+Trans" is that we use the GPT-2 as the encoder and decoder, and use the table linearization to indicate the cell position instead of the field-infusing method. More details about the table linearization can be found in (Chen et al., 2020a). And the vector $\boldsymbol{z}_{m,c}$ is fed to the last Transformer layer of GPT-2 instead of the first layer, which brings less impact on the pretrained GPT-2.

4.4 Generate-then-Select Paradigm

By sampling multiple latent variables $\boldsymbol{z}_c \sim p(\boldsymbol{z}_c)$, our model can generate multiple candidate sentences $\widetilde{\boldsymbol{Y}} = (\widetilde{\boldsymbol{y}}^1, \widetilde{\boldsymbol{y}}^2, ..., \widetilde{\boldsymbol{y}}^{N_c})$ for the table $\boldsymbol{x}$ where N_c is the number of generated sentences. We propose to find out the best sentence by a trained selector. The generator optimized with MLE may focus more on the token-level matching than the sentence-level consistency while the selector will focus on improving the sentence-level scores. Therefore, it can be considered as an alternative of reinforcement learning. The selector scores each candidate sentence by $s_i = S_\chi(\widetilde{\boldsymbol{y}}^i, \boldsymbol{x})$ where χ denotes the parameters of the selector network. Note that we are not designing a selector $s_i = S_\chi(\widetilde{\boldsymbol{y}}^i, \boldsymbol{y})$ because the reference sentence $\boldsymbol{y}$ is not available in practice.

Recent work has provided several selectors including parsing-based and NLI-based models (Chen et al., 2020c). We can directly use these selectors but we aims to develop a more general selector jointly considering surface-level fidelity and logical fidelity. We use a mix of BLEU-3 (Papineni et al., 2002) and NLI-Acc (Chen et al., 2020a) scores to supervise the selector. In the training stage of the selector, we can get the gold scores of each generated candidate with the referenced sentence $\boldsymbol{y}$ by $s_i^* = S^*(\widetilde{\boldsymbol{y}}^i, \boldsymbol{y})$. Then, we use BERT to encode $\boldsymbol{x}$ and $\boldsymbol{y}^i$ followed with the average pooling layers to produce $\boldsymbol{h}_s$ and $\boldsymbol{h}_s^i$. Finally, we score the table-sentence pair represented by $(\boldsymbol{h}_s, \boldsymbol{h}_s^i)$ as follows:

$$\begin{aligned} \boldsymbol{h}_f &= \boldsymbol{h}_s \oplus \boldsymbol{h}_s^i \oplus |\boldsymbol{h}_s - \boldsymbol{h}_s^i| \oplus \boldsymbol{h}_s \odot \boldsymbol{h}_s^i \\ S_\chi(\widetilde{\boldsymbol{y}}^i, \boldsymbol{x}) &= \sigma(\boldsymbol{W}_s \boldsymbol{h}_f) \end{aligned} \tag{12}$$

where $\oplus$ and $\odot$ denote the concatenation and the element-wise multiplication operations, respectively. And $\boldsymbol{W}_s$ denotes the parameters of the scoring network. The score $S_\chi(\widetilde{\boldsymbol{y}}^i, \boldsymbol{x})$ is between 0 and 1, and better sentences need to be closer to 1. The scores of gold reference are set to 1. Then we use the margin-based triplet loss for the generated sentences in two way: comparing with gold sentences, and comparing between arbitrary two generated sentences. Given N_c generated candidate sentences, we rank the generated sentences according to the mix of BLEU-3 and NLI-Acc scores. The ranked sentences are denoted by $\widetilde{\boldsymbol{Y}}_r = (\widetilde{\boldsymbol{y}}_r^1, \widetilde{\boldsymbol{y}}_r^2, ..., \widetilde{\boldsymbol{y}}_r^{N_c})$ where $\widetilde{\boldsymbol{y}}_r^1$ has the highest score. Then the loss is computed as

follows:

$$\mathcal{L}_\chi = \max\left(0, S_\chi(\widetilde{\boldsymbol{y}}_r^i, \boldsymbol{x}) - S(\boldsymbol{y}, \boldsymbol{x}) + \gamma_1\right) + \max\left(0, S_\chi(\widetilde{\boldsymbol{y}}_r^j, \boldsymbol{x}) - S(\widetilde{\boldsymbol{y}}_r^i, \boldsymbol{x}) + \gamma_2\right) \tag{13}$$

where γ_1 and γ_2 are the hyperparameters representing margin values, and i and j represent the ranked indexes. At the inference stage, we can select the best sentence with the highest score.

Dataset	Vocab	Tables	Sentences	Train / Val. / Test
LogicNLG	122K	7,392	37,015	28,450 / 4,260 / 4,305
Logic2Text	14K	5,554	10,753	8,566 / 1,095 / 1,092

Table 1: The statistics of two datasets.

5 Experiments

5.1 Datasets

We conduct experiments on two datasets: LogicNLG (Chen et al., 2020a) and Logic2Text (Chen et al., 2020e). LogicNLG is constructed based on the positive statements of the Tabfact dataset (Chen et al., 2020c), which contains rich logical inferences in the annotated statements. Logic2Text is a smaller dataset and provides the annotation of logical forms. Since the annotations of logical forms are labor-intensive, we only use the table-sentence pairs, following the task formulation of LogicNLG. The statistics of the two datasets are shown in Table 1.

5.2 Evaluation and Settings

The models are evaluated on the **surface-level consistency** and the **logical fidelity**. In terms of the surface-level consistency, we evaluate models on the sentence-level BLEU scores (Papineni et al., 2002) based on 1-3 grams matching. In terms of logical fidelity, we follow the recent work and apply three metrics including SP-Acc and NLI-Acc based on semantic parsing and pretrained NLI model, respectively (Chen et al., 2020a). The metrics are computed with the officially released codes[2].

Compared Models. We compare our models with both non-pretrained and pretrained models. The non-pretrained models include "Field-Gating" (Liu et al., 2018) and "Field-Infusing" (Lebret et al., 2016) with LSTM decoder or Transformer decoder, which are strong baselines among non-pretrained models. The pretrained models include "BERT-TabGen" and "GPT-TabGen" with the base size (Chen et al., 2020a). Moreover, for the LogicNLG dataset, we compare with a two-phrase approach denoted by "GPT-Coarse-to-Fine", which first generates a template and then generates the final sentence conditioning on the template (Chen et al., 2020a). For the variational models, we compare with the vanilla CVAE (Kingma et al., 2014) that approximates the prior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ to $p(\boldsymbol{z})$.

Hyperparameters. For the non-pretrained models, we set the dimension of LSTM or Transformer to 256. Our model is based on "Field-Infusing+Trans" which includes 3-layer Transformers in the encoder and decoder respectively. The posterior network $q_\phi(\boldsymbol{z}_c|\boldsymbol{y})$ contains a two-layer Transformer. For the pretrained models, we use the base version of BERT and GPT-2 which have an embedding size of 768. The KL loss is minimized with the annealing trick where the KL weight is set to 0 for 2 epochs and grows to 1.0 in another 5 epochs. The learning rate is initialized to set to 0.0001 and 0.000002 for non-pretrained and pretrained models, respectively. Each model is trained for 15 epochs. A special setting for our model is that we generate 10 candidate sentences for each table, and report the average performance and the best performance based on the selector, respectively. We set the hyperparameters $\gamma_1 = 0.2$ and $\gamma_2 = 0.2$ for the selector.

5.3 Main Results

Table 2 and 3 present the performance of our model as well the compared models on the surface-level consistency and the logical fidelity. As shown, without the selector, our model DCVED already outperforms the baseline models "Field-Infusing" and "GPT-TableGen" on both LogicNLG and Logic2Text datasets. Specifically, when compared with "Field-Infusing", our model increases the BLEU-3, SP-Acc, and NLI-Acc scores by 1.4, 3.7, and 3.9 points, respectively on the LogicNLG dataset, and 0.2, 2.4, and 2.8 points on the Logic2Text dataset. When cooperating with GPT-2, our model outperforms "GPT-TableGen" by 1.6, 2.2, and 5.2 points of BLEU-3, SP-Acc, and NLI-Acc scores, respectively on the LogicNLG dataset, and 0.2, 1.3, and 5.4 points on the Logic2Text dataset. Moreover, our model

[2]https://github.com/wenhuchen/LogicNLG

Model	Type	Surface-Level Fidelity			Logical Fidelity	
		BLEU-1	BLEU-2	BLEU-3	SP-Acc	NLI-Acc
Non-Pretrained Models						
Field-Gating + LSTM	-	42.3	19.5	6.9	38.0	56.8
Field-Gating + Trans	-	44.1	20.9	8.3	38.5	57.3
Field-Infusing + LSTM	-	43.1	19.7	7.1	38.6	57.1
Field-Infusing + Trans	-	43.7	20.9	8.4	38.9	57.3
CVAE + Field-Infusing + Trans	-	46.4	23.1	9.4	39.8	59.0
DCVED + Field-Infusing + Trans	-	46.2	22.9	9.8	**42.6**	61.2
DCVED + Field-Infusing + Trans	Trained Selector	**47.4**	**23.4**	**10.6**	42.1	**62.5**
DCVED + Field-Infusing + Trans	Oracle NLI-Acc ‡	45.0	22.2	9.0	41.7	86.8
DCVED + Field-Infusing + Trans	Oracle BLEU-3 ‡	55.2	32.9	15.9	41.8	60.3
Pretrained Models						
BERT-TabGen	-	47.8	26.3	11.9	42.2	68.1
GPT-TabGen	-	48.8	27.1	12.6	42.1	68.7
GPT-TabGen	Adv-Reg	45.8	23.1	9.6	40.9	68.5
GPT-TabGen	RL	45.1	23.6	9.1	43.1	67.7
GPT-Coarse-to-Fine	-	46.6	26.8	13.3	42.7	72.2
CVAE + GPT-TabGen	-	49.0	27.9	13.5	42.6	71.8
DCVED + GPT-TabGen	-	49.3	28.3	14.2	**44.3**	73.9
DCVED + GPT-TabGen	Trained Selector	**49.5**	**28.6**	**15.3**	43.9	**76.9**
DCVED + GPT-TabGen	Oracle NLI-Acc ‡	49.7	28.5	14.5	46.1	92.2
DCVED + GPT-TabGen	Oracle BLEU-3 ‡	59.7	38.0	22.1	45.0	74.2

Table 2: The experimental results of different models on the test split of **LogicNLG** dataset, where we split the table into non-pretrained and pretrained models. The **bold** represents the best scores. Adv-Reg and RL denote adversarial regularization and reinforcement learning, respectively. Oracle-x represents the upper bound of the generated sentences.

Model	Type	Surface-Level Fidelity			Logical Fidelity	
		BLEU-1	BLEU-2	BLEU-3	SP-Acc	NLI-Acc
Non-Pretrained Models						
Field-Infusing + Trans	-	37.7	21.0	10.5	38.5	42.4
CVAE + Field-Infusing + Trans	-	37.1	20.4	9.3	38.1	41.6
DCVED + Field-Infusing + Trans	-	38.8	21.6	10.7	**40.9**	45.2
DCVED + Field-Infusing + Trans	Trained Selector	**39.4**	**22.0**	**11.0**	40.4	**48.2**
DCVED + Field-Infusing + Trans	Oracle NLI-Acc ‡	38.5	21.5	10.9	41.3	72.5
DCVED + Field-Infusing + Trans	Oracle BLEU-3 ‡	45.6	29.0	16.7	40.8	44.7
Pretrained Models						
GPT-TabGen	-	46.5	30.9	19.9	42.4	66.5
CVAE + GPT-TabGen	-	46.2	30.8	19.7	41.0	67.8
DCVED + GPT-TabGen	-	46.4	31.2	20.1	43.7	71.9
DCVED + GPT-TabGen	Trained Selector	**48.9**	**32.7**	**21.4**	**43.9**	**73.8**
DCVED + GPT-TabGen	Oracle NLI-Acc ‡	46.5	31.2	20.1	43.8	89.9
DCVED + GPT-TabGen	Oracle BLEU-3 ‡	52.1	37.5	26.1	43.5	72.0

Table 3: The experimental results of different models on the test split of **Logic2Text** dataset, where we split the table into non-pretrained and pretrained models. The **bold** represents the best scores. Oracle-x represents the upper bound of the generated sentences.

also outperforms the recent SOTA model "GPT-Coarse-to-Fine" which increases the NLI-Acc score from 72.2 to 73.9 points on the Logic2Text dataset. When combining with the trained selector, our model further increases the NLI-Acc scores to 76.9 and 73.8 points on LogicNLG and Logic2Text datasets, respectively. We also show the upper bound of our model on BLEU and NLI-Acc scores. Assume that two optimum selectors have access to the ground-truth sentences, and

Dataset	Model	BLEU-3	SP-Acc	NLI-Acc
LogicNLG	CVAE	9.4	39.8	59.0
	DCVED (z_c)	9.0	40.8	60.3
	DCVED (z_c, c)	9.3	40.1	60.2
	DCVED (z_c, z_m, m)	**10.2**	41.8	60.6
	DCVED (Full)	9.8	**42.6**	**61.2**
Logic2Text	CVAE	9.3	38.1	41.6
	DCVED (z_c)	9.7	40.2	42.3
	DCVED (z_c, c)	9.6	39.4	43.5
	DCVED (z_c, z_m, m)	**11.2**	40.8	44.8
	DCVED (Full)	10.7	**40.9**	**45.2**

Table 4: The performances of ablated models as well as the full model on the two datasets.

would select the best sentence according to the BLEU-3 and NLI-Acc scores, respectively. As shown, a higher BLEU-3 score does not lead to a higher NLI-Acc score. Similarly, a higher NLI-Acc score does not yield a higher BLEU-3 score. The findings indicate that selecting candidates only by BLEU-3 or only by NLI-Acc is not enough. Instead, our trained selector comprehensively considers the BLEU-3 and NLI-Acc scores.

5.4 Ablation Study

To analyze which mechanisms are driving the improvements, we present an ablation study in Table 4. We show different ablated models with different combinations of $\boldsymbol{z}_c$, $\boldsymbol{z}_m$, $\boldsymbol{c}$ and $\boldsymbol{m}$. All these models are based on "Field-Infusing". Moreover, the vanilla CVAE is also compared, which can be considered as a baseline making both $\boldsymbol{z}_m$ and $\boldsymbol{z}_c$ independent from $\boldsymbol{x}$.

As shown, both the mediators and the confounders are influential. The full model achieve the best SP-Acc and NLI-Acc scores with slightly lower BLEU-3 scores than the ablated model, DCVED ($\boldsymbol{z}_c$, $\boldsymbol{z}_m$, $\boldsymbol{m}$). Eliminating $\boldsymbol{c}$ from the full model leads to a drop of NLI-Acc by 0.6 and 0.4 points on LogicNLG and Logic2Text, respectively. Further eliminating $\boldsymbol{z}_m$ and $\boldsymbol{m}$ leads to a drop of NLI-Acc by 0.9 and 2.9 points on LogicNLG and Logic2Text, respectively. An interesting finding is that DCVED ($\boldsymbol{z}_c$, $\boldsymbol{c}$) performs worse than DCVED ($\boldsymbol{z}_c$) on SP-Acc. The reason may be that predicting $\boldsymbol{c}$ from $\boldsymbol{z}_c$ without considering the mediators $\boldsymbol{z}_m$ may also lead to a bias, similar to CVAE. However, the ablated models all perform better than CVAE on SP-Acc and NLI-Acc.

5.5 Human Evaluation

Following recent work (Chen et al., 2020a), we also perform human evaluation on the fluency and logical fidelity. We randomly select 200 tables in the LogicNLG dataset, and generate one sentence per table for each model. Then we present the generated sentences to four raters without telling which model generates them. The raters are all post-graduate students majoring in computer science. We ask the raters to finish two binary-decision tasks: 1) whether a generated sentence is fluent; and 2) whether the fact of a generated sentence can be supported by the given table. We report the averaged results in Table 5, from which we can see that our model "DCVED + GPT-TabGen" mainly increases the logical fidelity over the baseline model "GPT-TabGen" from 19.1% to 25.8%. When cooperated with the trained selector and the oracle NLI selector, our model further increase the logical fidelity to 30.8% and 37.1%, respectively. It is worth noting that the NLI selector can be represented by the scorer $P_{NLI}(\widetilde{\boldsymbol{y}}, \boldsymbol{x})$, which does not require the ground-truth sentence $\boldsymbol{y}$ to be available (Chen et al., 2020a). It means that the setting of using the oracle NLI selector is acceptable.

	fluency %	logical fidelity %
GPT-TabGen	96.4	19.1
+ DCVED	98.3	25.8
+ DCVED + Trained Selector	**99.5**	30.8
+ DCVED + Oracle NLI Selector	98.0	**37.1**

Table 5: The results of human evaluation on the LogicNLG dataset.

5.6 Case Study

To directly see the effect of our model, we present a case study in Figure 2. Several GPT-2 based models generate sentences describing two tables in the LogicNLG test set. The underlined red words represent the facts contradicting the table. As shown, for the first table, CVAE generates the sentence "*The album was released in the United State 2 time*", where the correct entity should be "*the United Kingdom*" according to the table. Instead, our model DCVED acknowledges that "*The album was released in the United Kingdom 2 time*". Moreover, compared with those deterministic models like GPT-TableGen and GPT-Coarse-to-Fine, our model can generate sentences with different logical types. For the second table, we can see that many contradicting facts exist in recent models. For example, GPT-TableGen generates an incomplete sentence, which uses superla-

Case 1: Black Ice (Album)

country	date
Europe	17 october 2008
Australia	18 october 2008
United Kingdom	20 october 2008
United Kingdom	1 december 2008
United States	20 october 2008
Japan	22 october 2008
Germany	5 december 2008
Global (itunes)	19 november 2012

GPT-TableGen: The album was released in the United State.
GPT-Coarse-to-Fine: Black Ice was released in Germany and Japan.
CVAE: The album was released in the United State 2 time.
DCVED: The album was released in the **United Kingdom 2 time**.
DCVED: The album was released in the **United State before** the release of the album in **Japan**.

Case 2: Green Party of Canada

election	of candidates nominated
1984	60
1988	68
1993	79
1997	79
2000	111
2004	308
2006	308
2008	303

GPT-TableGen: The Green Party Of Canada had the highest number of Candidate Nominated.
GPT-Coarse-to-Fine: The Green Party Of Canada had 308 more Candidate Nominated than 1984.
CVAE: The Green Party Of Canada had the highest number Of Nomination in the 2000 Election.
DCVED: The Green Party Of Canada had the highest number Of Nomination **in 2004**.
DCVED: The Green Party Of Canada had **more Candidate Nominated in 2004 than in 2000**.

Figure 2: The case study of different GPT-2 based models for two tables in the LogicNLG test set. The underlined red words represent the facts not supported by the table. For our model DCVED, we present two generated sentences for each table.

tive logic but not mentions a specific year. Instead, our model produces two logically consistent sentences with superlative and comparative logic.

5.7 Limitations

Although our model can improve the logical fidelity to a certain degree, all the models still get low scores in terms of the logical fidelity in human evaluation, which reflects the challenge of the task. Especially, we find that models do not perform well on certain types of tables: 1) containing and comparing between large numbers, e.g., 18,013 and 29,001 in a table; and 2) containing mixed logics so that models require multi-hop reasoning, e.g., models generating "there were 3 nations that won 2 gold medals" while the correct nation number is 4.

To deal with these problems, we believe that two directions of work may be workable: 1) enhancing the mediators. For example, the logical forms (Chen et al., 2020e) can be utilized as the mediator. But as mentioned in Section 4.2, it is label-intensive to annotate the logical forms; 2) large-scale knowledge grounded pre-training, which may be a more promising way. This type of work utilized the existing knowledge graphs or crawled data from Wikipedia (Chen et al., 2020b) to help models better encode/represent non-linguistic inputs, such as the numbers, the time, or the scores in the tables.

6 Conclusion

In this paper, we propose a de-confounded variational encoder-decoder for the logical table-to-text generation. Firstly, we assume two latent variables existed in the continuous space, representing the mediator and the confounder respectively. And we apply the causal intervention method to reduce the spurious correlations. Secondly, to make the latent variables meaningful, we use the exactly selected entities to supervise the mediator and the not selected but linguistically similar entities to supervise the confounder. Finally, since our model can generate multiple candidates for a table, we train a selector guided by both surface-level and logical fidelity to select the best sentence. The experiments show that our model yields competitive results with recent SOTA models.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China under Grant 2018YFC0830400, and Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

References

Xuefeng Bai, Linfeng Song, and Yue Zhang. 2020a. Online back-parsing for amr-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1206–1219. Association for Computational Linguistics.

Yang Bai, Ziran Li, Ning Ding, Ying Shen, and Hai-Tao Zheng. 2020b. Infobox-to-text generation with

tree-like planning based attention network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3773–3779. ijcai.org.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7929–7942. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8635–8648. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020c. Tabfact: A large-scale dataset for table-based fact verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020d. Exploring logically dependent multi-task learning with causal inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2213–2225. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020e. Logic2text: High-fidelity natural language generation from logical forms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2096–2111. Association for Computational Linguistics.

Sebastian Gehrmann, Falcon Z. Dai, Henry Elder, and Alexander M. Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 46–56. Association for Computational Linguistics.

Unnat Jain, Ziyu Zhang, and Alexander G. Schwing. 2017. Creativity: Generating diverse questions using variational autoencoders. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5415–5424. IEEE Computer Society.

Katherine A. Keith, David Jensen, and Brendan O'Connor. 2020. Text and causal inference: A review of using text to remove confounding from causal estimates. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5332–5344. Association for Computational Linguistics.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1203–1213. The Association for Computational Linguistics.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4881–4888. AAAI Press.

Christos Louizos, Uri Shalit, Joris M. Mooij, David A. Sontag, Richard S. Zemel, and Max Welling. 2017. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6446–6456.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 11–19. The Association for Computer Linguistics.

Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2047–2057. Association for Computational Linguistics.

Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. 2018. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105(4):987–993.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4658–4664. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1173–1186. Association for Computational Linguistics.

Judea Pearl. 2010. On measurement bias in causal inference. *uncertainty in artificial intelligence*, pages 425–432.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7987–7998. Association for Computational Linguistics.

Jakub M. Tomczak and Max Welling. 2018. VAE with a vampprior. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *neural information processing systems*, pages 5998–6008.

Liwei Wang, Alexander G. Schwing, and Svetlana Lazebnik. 2017. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5756–5766.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1072–1086. Association for Computational Linguistics.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced amr-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 732–741. Association for Computational Linguistics.