香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Enhancing the Controllability and Quality of Text Generation

LIU Guangyi
刘广熠

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer and Information Engineering

The Chinese University of Hong Kong, Shenzhen

June 2023

<u>Thesis Assessment Committee</u>

Professor HUANG Rui (Chair)

Professor CUI Shuguang (Thesis Supervisor)

Professor LI Zhen (Thesis Co-supervisor)

Professor WANG Benyou (Committee Member)

Professor WANG Liwei (Examiner from CUHK)

Professor LIANG Xiaodan (External Examiner)

Abstract

of thesis entitled:

Enhancing the Controllability and Quality of Text Generation

Submitted by LIU Guangyi

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong, Shenzhen in June 2023

In the rapidly evolving field of natural language processing, the controllability and quality of text generation are fundamental to the development and application of robust, effective models. This thesis presents a comprehensive exploration of two novel methodologies designed to enhance these aspects: the Edit-Invariant Sequence Loss (EISL) and Composable Text Controls in Latent Space with Ordinary Differential Equations (ODEs).

The first section of the thesis delves into the EISL approach, a novel loss function that transcends the traditional sequence cross-entropy loss's token-by-token match paradigm. EISL's robustness to various noises and edits in the target sequences is particularly effective in handling imperfect target sequences, showing a significant improvement in tasks such as machine translation with noisy targets, unsupervised text style transfer, and non-autoregressive generation.

The second part explores the technique of Composable Text Controls in Latent Space using ODEs. This method offers an efficient and flexible way to compose a

broad range of text control operations in the compact latent space of text. The low-dimensionality and differentiability of the text latent vector allow the development of an efficient sampler, linking pretrained language models to the latent space, and decoding sampled vectors into desired text sequences.

By analyzing these two methodologies, this thesis underlines their combined potential to enhance both the controllability and quality of neural text generation. Experimental results evidence substantial improvement over traditional methods, thereby opening new perspectives for future research in the design of more efficient, flexible, and high-quality text generation systems.

摘要

提升文本生成的可控性和可组合性

在快速发展的自然语言处理领域中，文本生成的可控性和质量对于发展和应用强大有效的模型至关重要。本论文全面探讨了两种旨在提高这些方面的新颖方法：编辑不变序列损失（EISL）和在潜在空间中使用常微分方程（ODEs）进行可组合的文本控制。

论文的第一部分深入研究了 EISL 方法，这是一种超越了传统序列交叉熵损失逐个词符匹配范例的新型损失函数。EISL 对目标序列中的各种噪声和编辑的鲁棒性在处理不完美的目标序列时表现出色，如在处理带有噪声目标的机器翻译、无监督的文本风格转换和非自回归生成等任务中，显示出明显的改进。

第二部分探讨了在潜在空间中使用 ODEs 进行可组合的文本控制的技术。这种方法提供了一种在文本的紧凑潜在空间中组合广泛的文本控制操作的有效和灵活的方式。文本潜在向量的低维性和可微分性允许开发一个高效的采样器，将预训练的语言模型连接到潜在空间，并将采样的向量解码成期望的文本序列。

通过分析这两种方法，本论文强调了它们结合增强神经文本生成的可控性和质量的潜力。实验结果证明了相比于传统方法有显著的改进，从而为未来在设计更高效、灵活和高质量的文本生成系统的研究开启了新的视角。

Acknowledgement

I would like to express my deep appreciation and gratitude to my advisors, Prof. Shuguang Cui and Prof. Zhen Li, for their invaluable guidance, support, and encouragement throughout my graduate studies. Their unwavering commitment to my success has been truly inspiring and I am grateful for their mentorship.

I would also like to express my sincere gratitude to my collaborators, Prof. Zhiting Hu, Prof. Xiaodan Liang, and Dr. Zichao Yang, for their invaluable guidance, support, and constructive feedback throughout my research. Their expertise and insights have greatly enriched my work and have contributed significantly to the success of this thesis. I feel privileged to have had the opportunity to work with such distinguished scholars and am deeply grateful for their contributions and advice.

I am deeply grateful to my advisor, Prof. Eric Xing, at MBZUAI for his invaluable feedback and unwavering support during my tenure as a visiting student. His wealth of knowledge and experience have been instrumental in shaping the direction of my research and pushing me to achieve my full potential.

Furthermore, I would like to express my sincere gratitude to my internship advisors, Dr. Xiaodong He and Dr. Junwei Bao, for providing me with an exceptional opportunity to gain practical experience and apply my research skills in a professional setting. Their guidance and mentorship were instrumental in my professional development, and I am immensely grateful for the invaluable experience and knowledge that they imparted.

I owe a tremendous debt of gratitude to my laboratory colleagues, including Boyun Tan, Changyi Ma, Weibing Zhao, Yijue Dai, Yiwen Hu, Wending Zhou, Zhuo Li, Yinghong Liao, Xu Yan, Chaoda Zheng, Huijun Xing, and many others. Their camaraderie, support, and intellectual stimulation have been invaluable to me throughout my research journey. The collaborative environment in our lab has been a constant source of motivation and inspiration, and I feel incredibly fortunate to have worked alongside such a talented and dedicated group of individuals. I will always cherish the memories of our time together, including the days we spent working hard and pushing ourselves to our physical limits in the gym.

Finally, I would like to express my deepest gratitude to my wife for her unwavering support, encouragement, and patience throughout my graduate studies. Her love and understanding have been my constant source of strength.

Thank you all for your contributions to my academic and personal growth.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Significance

Machine learning, and more specifically, the area of text generation within Natural Language Processing (NLP), has seen widespread applicability across a variety of tasks such as machine translation, text summarization, and dialogue systems. This applicability has been made possible primarily due to the proliferation of models trained by maximizing the log-likelihood of the output sequence based on inputs using cross entropy (CE) loss. This method is efficient, easily implementable, and has been instrumental in building large-scale successful text generation models.

Despite the significant advancements, limitations persist. Conventional CE loss minimizes the negative log-likelihood of only the reference output sequence, penalizing all other sequences equally. This becomes restrictive as many plausible paraphrases close to a given reference sentence should not be treated as negative samples. Models trained with CE loss struggle to capture the invariance property of text, falling short when the supervision from a target sequence is imperfect due to noise or weak supervision.

Moreover, when it comes to text control operations - editing text with respect to various attributes, manipulating keywords, generating new text of diverse properties - current models often require finetuning for each specific combination of operations. This approach is unscalable due to the combinatorial nature of potential compositions and the lack of supervised data. Even recent attempts at plug-and-play solutions struggle with the complexity of search or optimization in text sequence space due to the discrete nature of text and the high-dimensionality of sequence space.

1.2 Research Objectives

This thesis aims to tackle the above challenges by introducing and integrating two novel methodologies: the Edit-Invariant Sequence Loss (EISL) and the Composable Text Controls in Latent Space. EISL proposes an alternative loss to CE that models the matching of each reference n-gram across all n-grams in a candidate sequence, effectively capturing the edit invariance properties of text n-grams and thereby improving the model's ability to handle noise and imperfect supervision in target sequences.

On the other hand, Composable Text Controls in Latent Space, implemented through an approach called LatentOps, offers a more efficient solution for diverse text control operations. It operates in the compact and continuous latent space of text, allowing for more efficient and accurate generation of high-quality text sequences.

By exploring these methodologies, the thesis aims to not only enhance the controllability and quality of the generated text but also contribute new insights that could inform future research and development in this domain.

1.3 Thesis Overview

The subsequent chapters of this thesis are structured as follows:

- Chapter 2, reviews prior studies that form the backdrop of our research, ranging from advancements in deep neural sequence models and text control in sequence and latent spaces, to learning with noisy labels in classification. This background is crucial to understand the existing challenges and the motivation behind the novel methodologies presented in the subsequent chapters.

- Chapter 3 presents the concept and application of the Edit-Invariant Sequence Loss (EISL). It explains its robustness to various noises and edits in the target sequences and its effectiveness in tasks like machine translation with noisy targets, unsupervised text style transfer, and non-autoregressive generation. Experimental results show that EISL loss can be easily incorporated with a series of sequence models and outperform Cross-Entropy and other popular baselines across the board.
- Chapter 4 delves into the methodology of Composable Text Controls in Latent Space using LatentOps. This approach provides an efficient and flexible way to compose a wide range of text control operations in the compact latent space of the text. The experimental results demonstrate that composing operators within our method manages to generate or edit high-quality text, substantially improving over respective baselines in terms of quality and efficiency.
- Chapter 5 concludes with a comprehensive summary of the findings, their implications for the field, and directions for future research.

In addition to the main content, this thesis includes an Appendix. This section provides additional experimental results and in-depth analyses that further illustrate and validate the effectiveness of the proposed methodologies.

Overall, this thesis serves as a comprehensive exploration of these novel methodologies and their potential to transform the landscape of text generation within machine learning.

1.4 Bibliographic Note

Portions of this thesis are based on prior peer-reviewed publications:

- **Chapter 3**: Guangyi Liu, Zichao Yang, Tianhua Tao, Xiaodan Liang, Junwei Bao, Zhen Li, Xiaodong He, Shuguang Cui, Zhiting Hu. ”Don’t Take It Literally: An Edit-Invariant Sequence Loss for Text Generation” [Liu et al., 2022b].

- **Chapter 4**: Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, Zhiting Hu. ”Composable Text Controls in Latent Space with ODEs” [Liu et al., 2022a].

The code for the techniques presented in this thesis are available at `https://github.com/guangyliu`

□ **End of chapter.**

Chapter 2

Related Work

In this chapter, we will dive into the existing studies that have paved the way for the advancements introduced in this thesis. These studies encompass deep learning methodologies for sequence models and text control techniques, all instrumental in the field of text generation.

2.1 Deep Neural Sequence Models

Deep neural sequence models, including recurrent neural networks [Sutskever et al., 2014; Mikolov et al., 2010] and transformers [Vaswani et al., 2017], have made considerable progress in various text generation tasks, such as machine translation [Bahdanau et al., 2015; Vaswani et al., 2017]. These models, generally trained with the maximum-likelihood objective, may exhibit sub-optimal performance due to cross-entropy's exact sequence matching assumption.

Numerous works have tried to address this issue. For instance, some studies [Ranzato et al., 2016; Rennie et al., 2017; Liu et al., 2017; Shen et al., 2016; Smith and Eisner, 2006] proposed using policy gradient or minimum risk training to optimize the expected BLEU metric [Papineni et al., 2002a]. However, these can lead to high variance and instability in reinforcement learning training. To combat this, soft Q-learning was introduced [Guo et al., 2021], and new reward functions based on semantic similarity for translation were developed [Wieting et al., 2019].

Moreover, initial attempts have been made to create differentiable BLEU objectives [Zhukov and Kretov, 2017; Casas et al., 2018] through soft approximations to the count of n-gram matching in the original BLEU formulation. Some researchers [Shao et al., 2018, 2021, 2020] have minimized the n-gram difference between model outputs and targets in non-autoregressive generation.

Research in learning with noisy labels in classification [Zhang and Sabuncu, 2018; Xu et al., 2019; Wang et al., 2019b; Hu et al., 2019] is also relevant to our work. In the context of text generation, student forcing has been proposed to substitute teacher forcing [Nicolai and Silfverberg, 2020], potentially mitigating the influence of noise in the target sequence during decoding. Another approach is loss truncation [Kang and Hashimoto, 2020], which adaptively removes high-loss examples considered as invalid data.

2.2 Text Control in Text Generation

Contemporary work on text generation can be broadly categorized into two: those generating desirable texts by directly modifying the text sequence space, and those operating on the latent space to obtain a representation that can be decoded into a sequence with desired attributes.

2.2.1 Text Control in Sequence Space

Studies involving large autoregressive language models like GPT-2 have demonstrated success in text generation, investigating conditional generation by conducting operations on the sequence space of these models. For instance, the plug-and-play framework [Dathathri et al., 2020] utilizes the gradients of attribute classifiers to modify the hidden states of the pretrained language model at each step. Other research, like FUDGE [Yang and Klein, 2021] and MUCOCO [Kumar et al., 2021], have introduced different approaches for modifying the sequence space.

2.2.2 Text Control in Latent Space

The other category includes methods that control text generation by modifying the text representation in the latent space. These methods typically utilize a Variational Autoencoder (VAE) to encode the input sequence into a latent representation [Mueller et al., 2017; Liu et al., 2020]. Then, attribute networks that are jointly trained with the VAE are used to obtain a modified representation that can be decoded into the desired sequence.

For instance, PPVAE [Duan et al., 2020] uses an unconditional Pre-train VAE and a conditional Plugin-VAE for this purpose. Plug and Play [Mai et al., 2020a] follows a similar framework but replaces the VAE with an Auto-encoder and the Plugin-VAE with an MLP to obtain the desired vector. Some methods employ an attribute classifier to edit the latent representation with Fast-Gradient-Iterative-Modification [Wang et al., 2019a]. Given the recent success of diffusion models, LDEBM [Yu et al., 2022] proposes a diffusion process in the latent space for text generation.

The research conducted in this thesis builds upon these existing foundations, aiming to contribute to the evolution of text generation in machine learning.

Chapter 3

EISL: An Edit-Invariant Sequence Loss for Text Generation

3.1 Introduction

Neural text generation models have ubiquitous applications in natural language processing, including machine translation [Bahdanau et al., 2015; Sutskever et al., 2014; Wu et al., 2016; Vaswani et al., 2017], summarizations [Nallapati et al., 2016; See et al., 2017], dialogue systems [Li et al., 2016], etc. They are typically trained by maximizing the log-likelihood of the output sequence conditioning on the inputs with the cross entropy (CE) loss. The CE loss can be easily factorized into individual loss terms and can be optimized efficiently with stochastic gradient descent. Due to its computational efficiency and ease to implement, the training paradigm has played an important role in building successful large text generation models [Lewis et al., 2020; Radford et al., 2019a].

However, the CE loss minimizes the negative log-likelihood of only the reference output sequence, while all other sequences are equally penalized through normalization. This is over-restrictive since for a given reference target sentence, many possible paraphrases are semantically close, hence should not completely be treated as negative samples. For example, as shown in Figure 3.1, `a cat is on the red blanket` should be treated equally with `on the red blanket there is a cat`. A model trained with CE loss falls short of modeling such type of invariance for text.

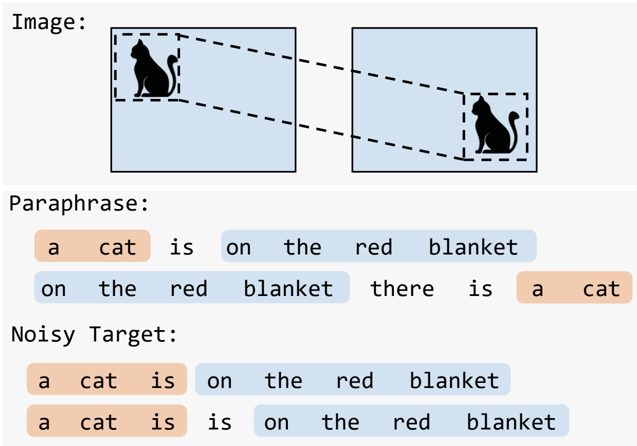

Figure 3.1: Invariance exists in both image and text, e.g., image is invariant to translation (top), and text is robust to many forms of edits (bottom).

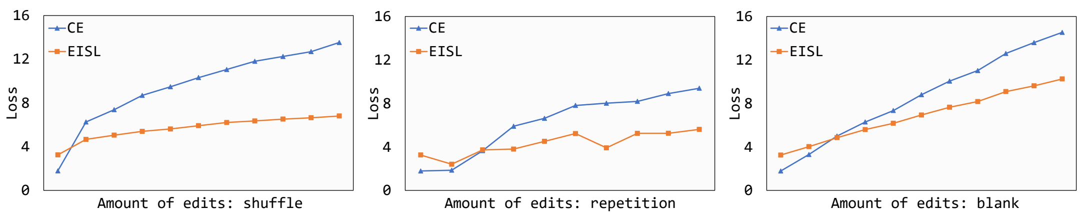

Figure 3.2: Sensitivity of CE and EISL loss w.r.t different types of text edits as the amount of edits increases (x-axis). We use a fixed machine translation model, synthesize different types of edits on target text, and measure the CE and EISL losses, respectively. The edit types include shuffle (changing the word order), repetition (words being selected are repeated), and word blank (words being replaced with a blank token). CE loss tends to increase drastically once a small amount of edits is applied. In contrast, EISL loss increases much more slowly, showing its robustness.

The problem is even exaggerated when the supervision from a target sequence is not perfect [Pinnis, 2018]. On one hand, there could be *noises* in the reference sequence which makes itself not a valid sentence. As in the last example shown in

Figure 3.1, there is a repetition error in the target sequence, which is common in human generated text. With the CE loss, the model is forced to copy all tokens including the error, and assign a high loss for the grammatically correct sequence. The exact tokens matching renders the CE loss sensitive to noises in the target, as shown in Figure 3.2. On the other hand, there are many problems with only *weak* supervision for target sequences [Tan et al., 2020; Wang et al., 2021; Lin et al., 2020]. For example, in tasks of unsupervised text style transfer [Jin et al., 2022] aiming to rewrite a sentence from one style to another, the original sentence offers weak supervision for the content (rather than the style). Yet using a CE loss here is problematic since it encourages the model to copy every original token.

Prior works have tried to address this problem using reinforcement learning (RL) [Guo et al., 2021; O'Neill and Bollegala, 2019; Wieting et al., 2019]. For example, policy gradient was used to optimize sequence rewards such as BLEU metric [Ranzato et al., 2016; Liu et al., 2017]. Such algorithms assign high rewards to sentences that are close to the target sentence. Though it is a valid objective to optimize, policy optimization faces significant challenges in practice. The high variance of gradient estimate makes the training extremely difficult, and almost all previous attempts rely on fine-tuning from models trained with CE loss, often with unclear improvement [Wu et al., 2018].

In this work, we propose an alternative loss to overcome the above weakness of CE loss, but reserve all nice properties such as being end-to-end differentiable, easy to implement, and efficient to compute, which hence can be used as a drop-in replacement or combined with CE. The loss is based on the observation that a viable candidate sequence shares many sub-sequences with the target. Our loss, called *edit-invariant sequence loss* (EISL), models the matching of each reference n-gram across all n-grams in a candidate sequence. The design is motivated by the translation invariance properties of ConvNets on images (see Figure 3.3), and captures the edit invariance properties of text n-grams in calculating the loss. Figure 3.2 shows the invariance property of EISL in comparison with CE. Appealingly, we show the conventional CE loss is a special case of EISL—when n equals to the sequence length, EISL calculates the exact sequence matching loss and reduces to CE. Moreover, the computations of EISL is essentially a convolution operation of candidate sequence using target n-grams as kernels, which is very easy to implement with existing deep learning libraries.

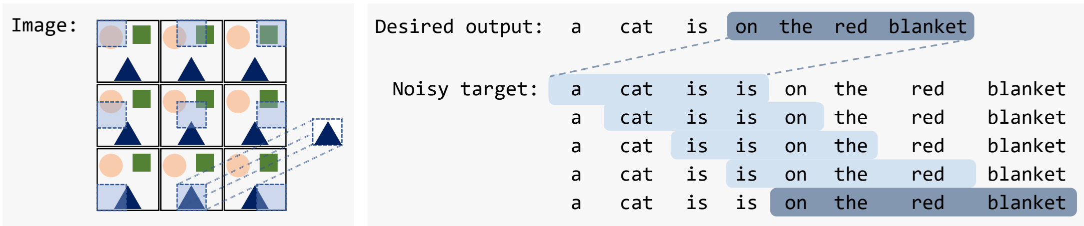

Figure 3.3: Inspired by the ConvNet convolution which applies a convolution kernel to different positions in an image and aggregate (**left**), we devise similar n-gram matching and convolution, which is robust to sequence edits (noises, shuffle, repetition, etc) (**right**).

To demonstrate the effectiveness of EISL loss, we conduct experiments on three representative tasks: machine translation with *noisy* training target, unsupervised text style transfer (only *weak* references are available), and non-autoregressive generation with *flexible generation order*. Experiments demonstrate EISL loss can be easily incorporated with a series of sequence models and outperforms CE and other popular baselines across the board.

3.2 Edit-Invariant Sequence Loss

In this section, we first review the conventional cross-entropy (CE) loss for sequence learning, and point out its weakness, especially when the target sequence is edited. We then introduce the EISL loss which gives a model the flexibility to learn from sub-sequences in a target sequence.

We first establish notations for the sequence generation setting. Let $(\boldsymbol{x}, \boldsymbol{y}^*)$ be a paired data sample where $\boldsymbol{x}$ is the input and $\boldsymbol{y}^* = (y_1^*, ..., y_{T^*}^*)$ is the reference target sequence. Define $\boldsymbol{y} = (y_1, ..., y_T)$ as a candidate sentence. Our goal is to build a model $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ that scores a candidate sequence $\boldsymbol{y}$ with parameter $\boldsymbol{\theta}$. In the sequel, we omit the condition $\boldsymbol{x}$ and the subscript $\boldsymbol{\theta}$ for simplicity.

3.2.1 The Difficulty of Cross Entropy Loss

The standard approach to learn the sequence model is to minimize the negative log-likelihood (NLL) of the target sequence, i.e., minimizing the CE loss $\mathcal{L}^{\text{CE}}(\boldsymbol{\theta}) = -\log p(\boldsymbol{y}^*)$. The CE loss assumes *exact* matching of a candidate sequence $\boldsymbol{y}$ with the target sequence $\boldsymbol{y}^*$. In other words, it maximizes the probability of only the target sequence $\boldsymbol{y}^*$ while penalizing all other possible sequence outputs that might be close but different with $\boldsymbol{y}^*$.

The assumption can be problematic in many practical scenarios: **(1)** For a given target sentence, there could be many ways of paraphrasing the sentence such as word reordering, synonyms replacement, active to passive rewriting, etc. Many of the paraphrases are viable candidate sequences, and/or share many sub-sequences with the reference sentence, and thus should not be treated completely as negative samples. Similar to the translation invariance which is shown to be effective in image modeling, a sequence loss that is *robust* to the shift and edits of sub-sequences in the reference sequence is preferred in order to model the rich variations of sequences; **(2)** The edit-invariance property is particularly desirable when the reference target sequence is corrupted with noise or is only weak sequence supervision. For instance, in Figure 3.3, the word `is` is repeated twice, which is one of the common errors in typing. Using CE loss in the noisy target setting forces the model to learn the data errors as well. In contrast, a sequence loss robust or invariant to the shift of sub-sequences assigns a high probability to the correct sentence even though it does not match the noisy target exactly. The loss thus offers flexibility for the model to select right information for learning.

3.2.2 EISL: Edit-Invariant Sequence Loss

Motivated by the above discussion, in this section, we draw inspirations from the convolution operation that enables translation invariance in image modeling (Figure 3.3, left), and propose an edit-invariant sequence loss (EISL) as illustrated in Figure 3.3 (right). Intuitively, for instance, given a 4-gram `on the red blanket`, because there is no extra knowledge to determine the position of the 4-gram in the noisy target sequence, we compute the losses across all positions in the noisy target sequence and aggregate. This is essentially a convolution over the target noisy sequence with the given n-gram as a convolution kernel.

We now derive the EISL loss in more details. Let $\boldsymbol{y}_{a:b} = (y_a, ..., y_{b-1})$ denote a sub-sequence of $\boldsymbol{y}$ that starts from index a and ends at index $b-1$, which is of length $b-a$. Thus $\boldsymbol{y}^*_{i:i+n}$ denotes the i-th n-gram in the reference $\boldsymbol{y}^*$. Denote $C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})$ as the number of times this n-gram occurs in $\boldsymbol{y}$:

$$C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y}) = \sum_{i'=1}^{T-n+1} \mathbb{1}(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}), \tag{3.1}$$

where $\mathbb{1}(\cdot)$ is the indicator function that takes value 1 if the n-grams match, and 0 otherwise. Intuitively, for a text generation model, we would like to maximize the occurrence of an n-gram from the reference in the target sequence. For a given probabilistic model $p_{\boldsymbol{\theta}}(\boldsymbol{y})$ (we omit the parameter $\boldsymbol{\theta}$ wherever the meaning is clear), the expected value of $C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})$ can be computed as follow:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}[C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})] &= \sum_{i'=1}^{T-n+1} \mathbb{E}_{p(\boldsymbol{y}_{i':i'+n})}\left[\mathbb{1}(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n})\right] \\ &= \sum_{i'=1}^{T-n+1} p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}).\end{aligned} \tag{3.2}$$

Thus, for each i-th n-gram in the reference, a straightforward way to define the learning objective is to minimize the negative log value of its expected occurrence, i.e., $-\log \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}[C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})]$.

The above loss requires computation of the marginal probability $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n})$ of an n-gram, which is intractable in practice. We therefore derive an upper bound of the loss and use it as the surrogate to minimize in training. We denote the upper bound surrogate as our EISL loss. Specifically, since for a given i', $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}) = \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'})p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}|\boldsymbol{y}_{<i'})$, then:

$$\begin{aligned}-\log \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}[C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})] &= -\log \sum_{i'=1}^{T-n+1} p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}), \\ &\leq \frac{-\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})} \sum_{i'=1}^{T-n+1} \log p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}|\boldsymbol{y}_{<i'})}{T-n+1} \\ &:= \mathcal{L}^{\text{EISL}}_{n,i}(\boldsymbol{\theta}).\end{aligned} \tag{3.3}$$

The detailed derivation is attached in Appendix 3.4.1. Notice that the EISL loss involves only the conditional distribution $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}|\boldsymbol{y}_{<i'})$ which is convenient to compute—we first sample tokens from the model up to the i' position, then compute NLL of the reference n-gram $\boldsymbol{y}^*_{i:i+n}$ occurring at position i' under the model distribution. The full n-gram EISL loss is then defined by averaging across all n-gram

positions in the reference:

$$\mathcal{L}_n^{\text{EISL}}(\boldsymbol{\theta}) = \frac{1}{T^* - n + 1} \sum_{i=1}^{T^*-n+1} \mathcal{L}_{n,i}^{\text{EISL}}(\boldsymbol{\theta}). \tag{3.4}$$

In practice, inspired by the standard BLEU metric (more in section 3.2.3), we could also straightforwardly combine different n-gram losses depending on tasks:

$$\mathcal{L}^{\text{EISL}}(\boldsymbol{\theta}) = \sum\nolimits_n w_n \cdot \mathcal{L}_n^{\text{EISL}}(\boldsymbol{\theta}), \tag{3.5}$$

where w_n is the weight of the n-gram loss. The rule of thumb is that a n-gram EISL loss with lower n is more robust to noises, as shown in our experiments. Following BLEU, we found that simply using equal weights for different n-grams up to $n = 4$ often produces good performance.

As discussed shortly, it is appealing that the n-gram EISL loss is indeed a direct generalization of the CE loss on the n-gram level: we sum the CE loss of an n-gram over all candidate sequence positions by conditioning on samples from the model. Besides, the derivation of the upper bound makes no assumption on the probability function $p(\boldsymbol{y})$, hence holds for both autogressive and non-autoregressive sequence models as demonstrated in our experiments.

Position Selection Minimizing the gram matching loss over all positions can make the model assign equal probabilities at all positions, which causes the training to collapse. We further adapt the loss to enable the model to automatically learn the positions of reference n-grams. For notation simplicity, let $g_{i,i'}^n$ denote the conditional probability $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'})$ involved above (Eq.3.3). We can vectorize the probability to get $\boldsymbol{g}_i^n = [g_{i,1}^n, ..., g_{i,T-n+1}^n]^T$, spanning all potential positions in the candidate sequence. We then normalize the probability vector $\boldsymbol{g}_i^n$ by Gumbel softmax [Jang et al., 2017], denoted as $\boldsymbol{q}_i^n = \texttt{Gumbel\_softmax}(\boldsymbol{g}_i^n)$, which we use as the weight for every n-gram positions. We multiply the weight with the original log probability

to get the new adjusted loss:

$$\mathcal{L}^{\text{EISL}}_{n,i}(\boldsymbol{\theta}) \approx -\boldsymbol{q}^n_i \cdot \log \boldsymbol{g}^n_i. \tag{3.6}$$

The loss can roughly be viewed as the "entropy" of the unnormalized probabilities $\boldsymbol{g}^n_i$, which has minimal value if the mass of the probability is assigned to one location only. Intuitively, if an $g^n_{i,i'}$ is large, then it is likely i' is the correct position for the reference n-gram, hence the weight for this position should also be large. This is like the greedy exploitation in reinforcement learning [Mnih et al., 2015]. On the other hand, to overcome over-exploitation, the Gumbel softmax introduces randomness in the weight assignment, which helps balance the exploitation-exploration trade-off in position selection for the model.

Efficient Approximate Computation: EISL as Convolution We show the EISL loss can be computed efficiently using the common convolution operator, with very little additional cost compared with the CE loss. The computation involves moderate approximation if the generation model is an autoregressive model, and is exact in the case of a non-autoregressive model (e.g., as in section 3.3.3). We first discuss the easy case when the model is a non-autoregressive model, where we have $g^n_{i,i'} = p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}) = \prod_{j=1}^n p(y_{i'+j-1} = y^*_{i+j-1})$. Denote V as the vocabulary size. Let $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, ...\boldsymbol{p}_T]$ be the probability output by the model across positions, where $\boldsymbol{p}_{i'} \in \mathbb{R}^V$ is the probability output after softmax at i'-th position, and each $\boldsymbol{p}_{i'}$ is independent with each other. On this basis, we compute the key quantity $\log \boldsymbol{g}^n_i$ in Eq. 3.6 as the direct output of the convolution operator.

As shown in Figure 3.4, we can get $\log \boldsymbol{g}^n_i$ by applying convolution on $\log \boldsymbol{P}$, with $\boldsymbol{y}_{i:i+n}$ as the kernels:

$$\log \boldsymbol{g}^n_i = \texttt{Conv}(\log \boldsymbol{P}, \texttt{Onehot}(\boldsymbol{y}^*_{i:i+n})), \tag{3.7}$$

where $\texttt{Onehot}(\cdot)$ maps each token to its corresponding one-hot representation and $\texttt{Conv}(\cdot,\cdot)$ is the convolution operation with the first argument as input and the second as the kernel. We transform $\boldsymbol{P}$ into log domain to turn the probability multiplication into log probability summations, where Conv can be directly applied. As shown in Figure 3.4, $\log \boldsymbol{P}$ is of shape $V \times T$ and $\texttt{Onehot}(\boldsymbol{y}^*_{i:i+n})$ is of shape $V \times n$, so $\texttt{Conv}(\log \boldsymbol{P}, \texttt{Onehot}(\boldsymbol{y}^*_{i:i+n}))$ is an one-dimensional convolution on the sequence axis. Formally, the i'-th convolutional output is:

$$\begin{aligned}\log g^n_{i,i'} &= \sum_{j=1}^{n} \log \boldsymbol{p}_{i'+j-1} \cdot \texttt{Onehot}(y^*_{i+j-1}) \\ &= \sum_{j=1}^{n} \log p(y_{i'+j-1} = y^*_{i+j-1} | \boldsymbol{y}_{<i'+j-1})\end{aligned} \tag{3.8}$$

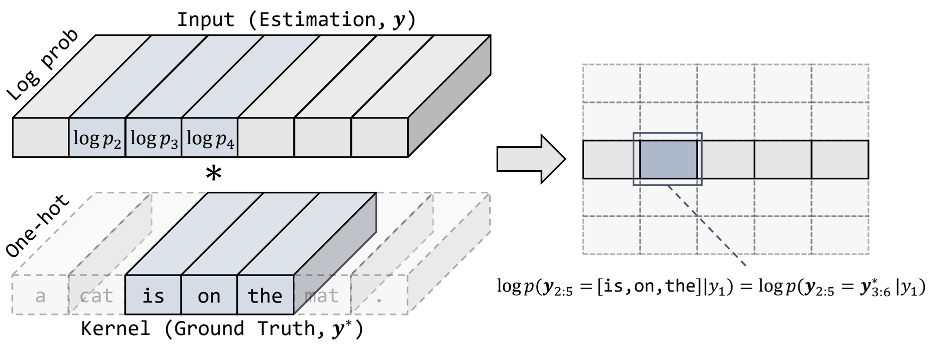

Figure 3.4: As convolution is a common operation for translation invariance in image, we adopt a convolution to achieve the translation invariance in text. The input is the distribution from the model output in log domain, kernel represents the convolution kernel and $*$ is the convolution operation. In this 3-gram example, there are 5 kernels, which correspond to the 5 rows on the right.

After obtaining $\boldsymbol{g}^n_i$ by convolution, the EISL loss in Eq. 3.6 can be easily calculated. We now discuss the case of autoregressive model, where by definition we have $g^n_{i,i'} = \prod_{j=1}^{n} p(y_{i'+j-1} = y^*_{i+j-1} | \boldsymbol{y}_{<i'}, \boldsymbol{y}^*_{i:i+j-1})$. The dependence on both $\boldsymbol{y}_{<i'}$ and $\boldsymbol{y}^*_{i:i+j-1}$ in each conditional makes exact estimation of $\log \boldsymbol{g}^n_i$ very complicated and costly. We thus introduce the approximation where we approximate $g^n_{i,i'}$ as

$\widetilde{g}^n_{i,i'} = \prod_{j=1}^n p(y_{i'+j-1} = y^*_{i+j-1} | \boldsymbol{y}_{<i'+j-1})$. That is, instead of conditioning on $\boldsymbol{y}^*_{i:i+j-1}$, we use the model-generated tokens $\boldsymbol{y}_{i':i'+j-1}$ as the condition. This simple approximation enables us to define the probability output $\boldsymbol{P}$ as in the non-autoregressive case, by just performing a forward pass of the model (i.e., sampling a token $\boldsymbol{y}'_i$ for each position i' and feeding it to the next step to get $\boldsymbol{p}_{i'+1}$). We can then apply the same convolution operator to approximately obtain $\log \boldsymbol{g}^n_i$ as in Eq. 3.7. Besides the great gain of computational efficiency, we note that the approximation is also effective, especially due to the *position selection* discussed above. Specifically, for each reference n-gram $\boldsymbol{y}^*_{i:i+n}$, the position selection in effect (softly) picks those large-value $g^n_{i,i'}$ (while dropping other low-value ones) to evaluate the loss. A large $g^n_{i,i'}$ value indicates the candidate $\boldsymbol{y}_{i':i'+n}$ is highly likely to match the reference $\boldsymbol{y}^*_{i:i+n}$, meaning that using $\boldsymbol{y}_{i':i'+n}$ in replacement of $\boldsymbol{y}^*_{i:i+n}$ is a reasonable approximation for evaluating the above conditionals. We provide empirical analysis of the approximation in Appendix 3.4.8, where we show the efficient approximate EISL loss values are very close to the exact EISL values.

3.2.3 Connections with Common Techniques

CE is a special case of EISL A nice property of EISL is that it subsumes the standard CE loss as a special case. To see this, set $n = T^*$ (the target sequence length), and we have:

$$\mathcal{L}^{\text{EISL}}_{T^*} = \mathcal{L}^{\text{EISL}}_{T^*,1} = -\log \boldsymbol{g}_1^{T^*} = -\log p(\boldsymbol{y} = \boldsymbol{y}^*) = \mathcal{L}^{\text{CE}}.$$

The connection shows the generality of EISL. As a generalization of CE, it enables learning at arbitrary n-gram granularity.

Connections between BLEU and EISL Both our method and the popular BLEU [Papineni et al., 2002b] metric use n-grams as the basis in formulation. Here we articulate the connections and difference between the two. Let us first take a review of the BLEU metric. Specifically, BLEU is defined as a weighted geometric mean of n-gram precisions:

$$
\begin{aligned}
\mathrm{BLEU} &= \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log \mathrm{prec}_n\right) \\
\mathrm{prec}_n &= \frac{\sum_{s \in \mathrm{gram}_n(\boldsymbol{y})} \min(C(\boldsymbol{s}, \boldsymbol{y}), C(\boldsymbol{s}, \boldsymbol{y}^*))}{\sum_{s \in \mathrm{gram}_n(\boldsymbol{y})} C(\boldsymbol{s}, \boldsymbol{y})},
\end{aligned}
$$

where BP is a brevity penalty depending on the lengths of $\boldsymbol{y}$ and $\boldsymbol{y}^*$; N is the maximum n-gram order (typically $N = 4$); $\{w_n\}$ are the weights which usually take $1/N$; prec_n is the n-gram precision, $\mathrm{gram}_n(\boldsymbol{y})$ is the set of unique n-gram sub-sequences of $\boldsymbol{y}$; and $C(\boldsymbol{s}, \boldsymbol{y})$ is the number of times a gram s occurs in $\boldsymbol{y}$ as defined in Eq. 3.1. The conventional formulation above enumerates over unique n-grams in $\boldsymbol{y}$. In contrast, we enumerate over token indexes in calculating the n-gram matching loss. BLEU considers the n-gram precisions and has a penalty term while EISL simply maximizes the log probability of n-gram matchings.

The non-differentiability of BLEU makes it hard to optimize directly, hence most prior attempts resort to reinforcement learning algorithms and use BLEU as the reward [Ranzato et al., 2016; Liu et al., 2017]. There are also some works trying to introduce differentiable BLEU metric using approximation like [Zhukov and Kretov, 2017]. However, such losses are often too complicated and have not yet demonstrated to perform well in practice.

3.3 Experiments

In this section, we present the experimental results on three text generation settings to test EISL's effectiveness, including learning from noisy text, learning from weak sequence supervision, and non-autoregressive generation models that require flexibility in generation orders. More details of the experimental setting are provided in Appendix 3.4.2.

3.3.1 Learning from Noisy Text

To test the robustness to noise, we evaluate on the task of machine translation with noisy training target, in which we train the models with noisy sequence targets and evaluate with clean test data.

Setup We test EISL loss on Multi30k and WMT18 raw corpus. We use German-to-English (de-en) dataset from Multi30k [Elliott et al., 2016], which contains 29k training instances. As inspired by Shen et al. [2019], to simulate various noises in the real data, we introduce four types of noises: shuffle, repetition, blank, and the synthetical noise, i.e., the combination of the aforementioned three types of noise. The noises are only added to the training target sequences. To verify the validity of EISL on real noisy data, we also use German-to-English (de-en) dataset from WMT18 raw corpus, which is a very noisy de-en corpus crawled from the web. We randomly select different number of training samples to test the influence of the data scale. We use a Transformer-based pretrained model BART-base [Lewis et al., 2020] and adopt greedy decoding in training and beam search (beam size $= 5$) in evaluation. We compare EISL loss with CE loss, Policy Gradient (PG), and Loss Truncation (LT). We also conduct ablation experiments to explore the effect of different n-grams in EISL loss. We use both BLEU [Papineni et al., 2002b] and BLEURT, an advanced model-based metric [Sellam et al., 2020], as the automatic metrics for evaluation. Due to space limit, we report BLEU results in the main paper, and defer BLEURT results in the appendix, where we can see BLEURT leads to the same conclusion as BLEU.

Results The results on noisy Multi30k are presented in Figure 3.5. The proposed EISL loss provides significantly better performance than CE loss and PG on all the noise types, especially on the high-level noise end. For synthetical noise as shown in Figure 3.5(d), it's interesting to see that CE and PG completely fail when the noise level is beyond 6, but model trained with EISL has high BLEU score, demonstrating EISL can select useful information to learn despite high noise. This validates that the proposed EISL is much less sensitive to the noise than the traditional CE loss and policy gradient training method. The results of different n-gram are shown in Figure 3.5(e). As the noise increases, the importance of lower grams, e.g., 1-gram, is more obvious. The results on real noisy data, WMT18 raw data, are shown in Figure 3.6. EISL loss achieves better performance than CE loss and PG, and the difference is getting larger when the training data scale increases. This again demonstrates EISL could learn more valid information in rather noisy data, while CE loss which only considers whole-sentence matching could struggle on noisy data. In Appendix 3.4.3, we provide more results (e.g., comparison with loss truncation [Kang and Hashimoto, 2020]) and case studies.

3.3.2 Learning from Weak Supervisions: Style Transfer

We experiment on transferring two types of text styles [Jin et al., 2022], namely sentiment and political slant, to verify EISL can learn from weak sequence supervisions.

Setup We use the Yelp review dataset and political dataset. Yelp contains almost 250k negative sentences and 380K positive sentences, of which the ratio of training, valid and test is $7:1:2$. Li et al. [2018a] annotated 1000 sentences as ground truth for better evaluation. The political dataset is comprised of top-level comments on Facebook posts from all 412 members of the United States Senate and House who

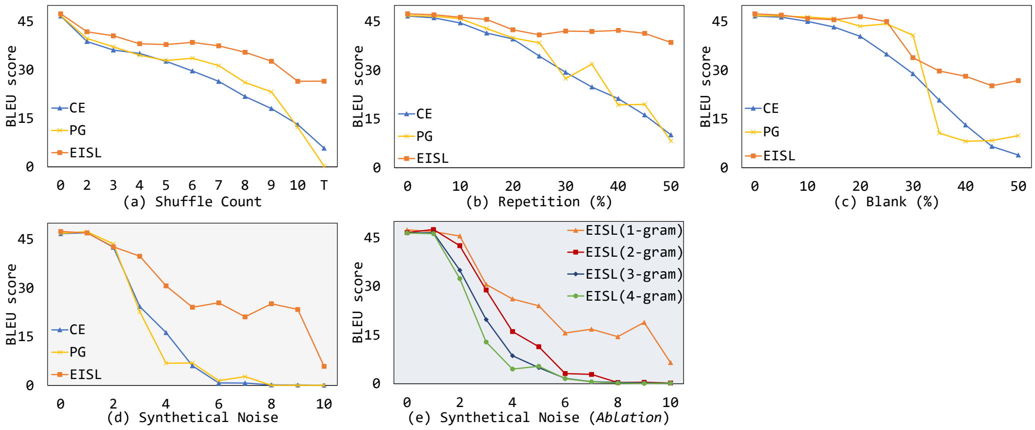

Figure 3.5: Results of Translation with Noisy Target on German-to-English(de-en) from Multi30k. BLEU scores are computed against clean test data. The x-axis of all figures denotes the level of noise we injected to target sequences in training. (a) Shuffle: selected tokens are shuffled; (b) Repetition: selected tokens are repeated; (c) Blank: selected tokens are substituted with a special blank token; (d) Synthetical noise: the combination of all three noises ($x = x_0$ stands for the combination of $5x_0\%$ of all kinds of noises); (e) Ablation study of n-grams for EISL on synthetical noise. BLEURT results are shown in Appendix 3.4.3.

have public Facebook pages [Voigt et al., 2018]. The data set contains 270K democratic sentences and 270K republican sentences. And there exists no ground truth for evaluation. The data preprocessing follows Tian et al. [2018]. The structured content preserving model [Tian et al., 2018] is adopted as the base model.

Following previous work, we compute automatic evaluation metrics: accuracy, BLEU score, perplexity (PPL) and POS distance. We also perform human evaluations on Yelp data to further test the transfer quality.

Results As sentiment results are shown in Table 3.1, the BLEU gets improved from 65.71 to 68.51 with EISL loss. On the premise of the correctness of sentiment transfer, EISL loss plays a critical role to guarantee lexical preservation. In the meanwhile, all of BLEU(human), PPL, and POS distance get improved. It is not surprising that EISL loss helps generate sentences more fluently and select the more appropriate words conditions on the content information. As the human evaluation results

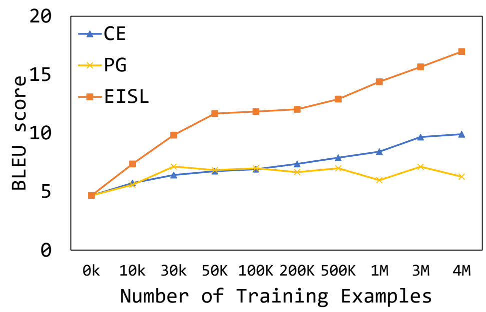

Figure 3.6: Results of German-to-English(de-en) Translation on WMT18 raw corpus. BLEU scores are computed against clean parallel test data. On x-axis, 0k denotes the performance of the pretrained model. BLEURT results are similar as shown in Appendix 3.4.3.

are shown in Table 3.1, the model with EISL loss performs better, in accord with the automatic metrics. After analyzing the generated samples, we found EISL loss could drive the model to adopt the words which fit the scene better and could understand more semantics but not just replace some keywords. See some examples in the Appendix 3.4.4.

We report the results of political data in Appendix 3.4.4. Our method outperforms all models on BLEU, PPL, and POS distance with comparable accuracy. For a more fair comparison with the base model, our EISL loss improves the base model on all four metrics, including the accuracy.

The results demonstrate the effectiveness of EISL for weak supervision task, improving both transfer accuracy fluency and content preservation.

3.3.3 Learning Non-Autoregressive Generation

Non-autoregressive neural machine translation (NAT, [Gu et al., 2018]) is proposed to predict tokens simultaneously in a single decoding step, which aims at reducing the inference latency. The non-autoregressive nature makes it extremely hard for models to keep the order of words in the sentences, hence CE often struggles with NAT

Model	Acc (%)	BLEU	BLEU (Human)	PPL	POS Distance
Hu et al. [2017a]	86.7	58.4	-	177.7	-
Shen et al. [2017]	73.9	20.7	7.8	72.0	-
He et al. [2020]	87.9	48.4	18.7	**31.7**	-
Dai et al. [2019a]	87.7	54.9	20.3	73.0	-
Tian et al. [2018]	**88.8**	65.71	22.56	42.07	0.352
with EISL (Ours)	**88.8**	**68.51**	**23.17**	41.56	**0.275**

Tian et al. [2018] (%)	*with* EISL (Ours) (%)	equal (%)
22.0	30.7	47.3

Table 3.1: **Top:** automatic evaluations on the Yelp review dataset. The BLEU (human) is calculated using the 1000 human annotated sentences as ground truth from Li et al. [2018a]. The first four results are from the original papers. **Bottom:** human evaluation statistics of base model vs. *with* EISL. The results denotes the percentages of inputs for which the model has better transferred sentences than other model.

problems. In experiments, we show EISL is superior to CE in NAT which requires modeling flexible generation order of the text.

Setup We use English-to-German dataset from WMT14 [Luong et al., 2015], which contains 4.5M training instances. We apply our proposed EISL loss on both fully NAT models [Gu et al., 2018; Sun et al., 2019] and iterative NAT models [Lee et al., 2018; Gu et al., 2019; Ghazvininejad et al., 2019], showing its general applicability and superiority, and we also compare with a wide range of recent methods [Shao et al., 2020; Wang et al., 2019c; Li et al., 2019; Ghazvininejad et al., 2020]. We evaluate with both BLEU and BLEURT metrics.

Results We first summarize the comparison of BLEU between EISL loss and CE loss in Table 3.2 (comparison of BLEURT is in Appendix 3.4.5). The proposed EISL improves the model performance on both the KD and original datasets. More specifically, for fully NAT models (Vanilla-NAT and NAT-CRF), EISL gives strong improvement. For iterative NAT models (iNAT, LevT, and CMLM), EISL also significantly outperforms the baselines when the iteration step is restricted to a small level as suggested by Kasai et al. [2020]. (We show in Appendix 3.4.5 that, with increasing iteration steps, the difference fades away. However, as studied in Kasai et al. [2020], iterative NAT models with many iteration steps do not hold the intrinsic advantage of speed since Transformer baselines with a shallow decoder can achieve comparable speedup and only at the sacrifice of minor performance drop.) Table 3.3 provides more comparison of with recent strong baselines. Specifically, we apply our EISL on the CMLM base model [Ghazvininejad et al., 2019] which shows strong superiority. We provide qualitative analysis in Appendix 3.4.5.

Decoding method	**Model**	**WMT14 en-de KD**		**WMT14 en-de**	
		CE	EISL	CE	EISL
Autoregressive	Transformer base [Vaswani et al., 2017]	27.48			
Non-Autoregressive	Vanilla-NAT [Gu et al., 2018]	17.9	**22.2**	9.12	**15.46**
	NAT-CRF [Sun et al., 2019]	21.88	**22.43**	-	-
	iNAT [Lee et al., 2018]	16.67	**22.59**	-	-
	LevT [Gu et al., 2019]	17.84	**23.61**	9.91	**18.47**
	CMLM [Ghazvininejad et al., 2019]	17.12	**23.05**	-	-

Table 3.2: The test-set BLEU of EISL loss and CE loss applied to non-autoregressive models. "KD" refers to the standard "knowledge distillation" setting in NAT [Gu et al., 2018]. iNAT, LevT and CMLM are iterative non-autoregressive models, that could run in multiple decoding iterations. However, the first decoding iteration of these models is fully non-autoregressive, which is what we use as our baselines.

Fully Non-Autoregressive model	WMT14 en-de KD
CMLM *with* CE [Ghazvininejad et al., 2019]	17.12
Auxiliary Regularization [Wang et al., 2019c]	20.65
Bag-of-ngrams Loss [Shao et al., 2020]	20.90
Hint-based Training [Li et al., 2019]	21.11
CMLM *with* AXE [Ghazvininejad et al., 2020]	23.53
CMLM *with* EISL (**Ours**)	**24.17**

Table 3.3: The test-set BLEU of CMLM trained with our EISL, compared to other recent fully non-autoregressive methods. The baseline results are from [Ghazvininejad et al., 2020], where CMLM-with-AXE generates 5 candidates and ranks with loss. Our method follows the same generation configuration as CMLM-with-AXE.

3.4 In-depth Derivation and Comprehensive Results

3.4.1 Detailed Derivation

For a given i',

$$
\begin{aligned}
&p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}) \\
&\quad = \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),
\end{aligned}
$$

then we derive the detail of Eq. 3.3 in Eq. 3.9, where the first inequality holds since $T - n + 1 \geq 0$; and the second inequality holds by Jensen's inequality.

3.4.2 Detailed Experimental Setup

Learning from Noisy Text

We use a Transformer-based pretrained model BART-base [Lewis et al., 2020], containing 6 layers in the encoder and decoder. We train the model using the Adam optimizer with learning rate 3×10^{-5} with polynomial decay and the maximum number of tokens is 6000 in one step. The models are trained on one Tesla V100 DGXS with

$$
\begin{aligned}
l_{n,i}^{\mathrm{EISL}}(\boldsymbol{\theta}) &= -\log \sum_{i'=1}^{T-n+1} p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}), \qquad (3.9)\\
&= -\log \frac{1}{T-n+1} \sum_{i'=1}^{T-n+1} \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}) - \log(T-n+1),\\
&\leq -\log \frac{1}{T-n+1} \sum_{i'=1}^{T-n+1} \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),\\
&\leq -\frac{1}{T-n+1} \sum_{i'=1}^{T-n+1} \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) \log p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),\\
&= -\frac{1}{T-n+1} \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})} \sum_{i'=1}^{T-n+1} \log p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),\\
&= \mathcal{L}_{n,i}^{\mathrm{EISL}}(\boldsymbol{\theta}),
\end{aligned}
$$

32GB memory. We start with CE training using teacher forcing for fast initialization. We then switch to combined 1- and 2-gram EISL with weight $0.8 : 0.2$, which we select using the validation set. We adopt greedy decoding in training and beam search (beam size $= 5$) in evaluation. We use fairseq[1] [Ott et al., 2019] to conduct the experiments. We compare EISL loss with CE loss and Policy Gradient (PG), where PG is used to finetune the best CE model. Teacher forcing is employed in CE training.

Learning from Weak Supervisions: Style Transfer

We use the Adam optimizer with learning rate 5×10^{-4}, the batch size is 128 and the model is trained on one Tesla V100 DGXS 32GB. We compare the results between the base model and the model with EISL. Specifically, on top of the base model, we add the EISL loss (a combination of 2, 3 and 4-gram with the same weights $1/3$) to reduce the discrepancy between the transferred sentence generated by language model and the original sentence. We assign EISL loss with weight 0.5.

[1] Fairseq(-py) is MIT-licensed.

Following previous work, we compute automatic evaluation metrics: accuracy, BLEU score, perplexity (PPL) and POS distance. For accuracy, we adopt a CNN-based classifier, trained on the same training data, to evaluate whether the generated sentence possesses the target style. Then we measure BLEU score and BLEU(human) score of transferred sentences against the original sentences and ground truth, respectively. PPL metric is evaluated by GPT-2 [Radford et al., 2019a] base model after finetuning on the corresponding dataset, with the goal to assess the fluency of the generated sentence. POS distance is used to measure the model's semantics preserving ability [Tian et al., 2018].

We also perform human evaluations on Yelp data to further test the transfer quality. We first randomly select 100 sentences from the test set, use these sentences as input and generate sentences from the base model [Tian et al., 2018] and our model. Then for each original sentence, we present the outputs of the base model and ours in random order. The three annotators are asked to evaluate which sentence is preferred as the transferred sentence of the original sentence, in terms of content preservation and sentiment transfer. They can choose either output or select the same quality. We measure the percentage of times each model outperforms the other.

Learning Non-Autoregressive Generation

We use the Adam optimizer with learning rate 5×10^{-4} with inverse square root scheduler. We apply sequence-level knowledge distillation to the dataset, which can reduce the complexity of the dataset, making it easier for the model to learn and improving the performance. The models are first trained by CE loss for fast initialization, then focus on 2-gram, 3-gram, and 4-gram with the same weights. Fairseq [Ott et al., 2019] is adopted to conduct the experiments. We average the last 5 checkpoints as the final model.

3.4.3 Additional Results of Learning from Noisy Text

Results of BLEURT Metric

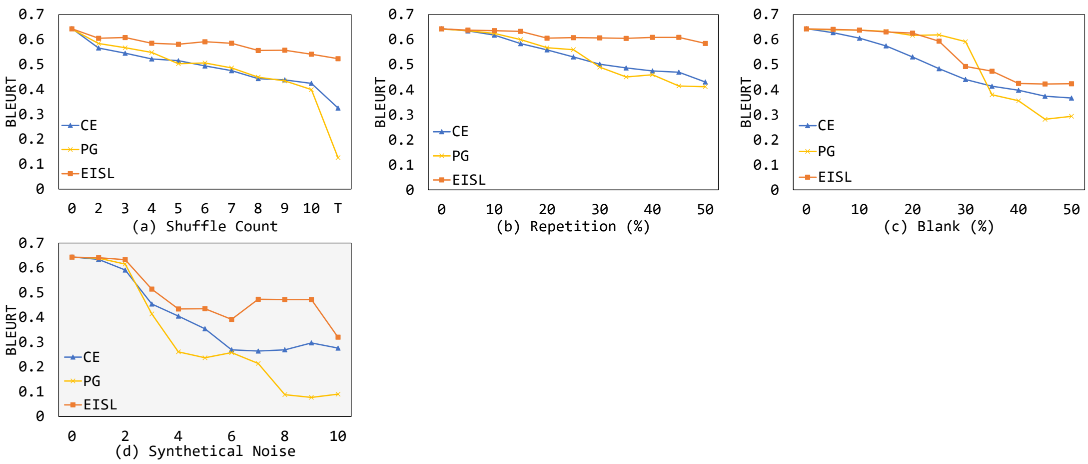

Figure 3.7: Results of Translation with Noisy Target on German-to-English(de-en) from Multi30k. BLEURT scores are computed against clean test data.

In this section, we evaluate the results of CE, PG and EISL on BLEURT [Sellam et al., 2020] metric. We use the recommended BLEURT-20 checkpoint. It gives a score for every sentence pair, and we averaged the scores to get the final score. The results are shown in Figure 3.7. Both BLEU metric and BLEURT metric show the superiority of our proposed EISL loss.

Comparison with Loss Truncation

The Loss Truncation (LT [Kang and Hashimoto, 2020]), method adaptively removes high log loss examples as a way to optimize for distinguishability. In this section, We'd like to show the comparisons with Loss Truncation. We evaluated two variants of LT: (1) LT_Pre which first trains the model with CE loss and then adds LT for further training, and (2) LT which directly trains the model with CE loss and LT together. Hyperparameters were selected on the validation set. For simplicity, we remove the PG curves (Figure 3.5), and the comparison results with LT are shown in Figure 3.8.

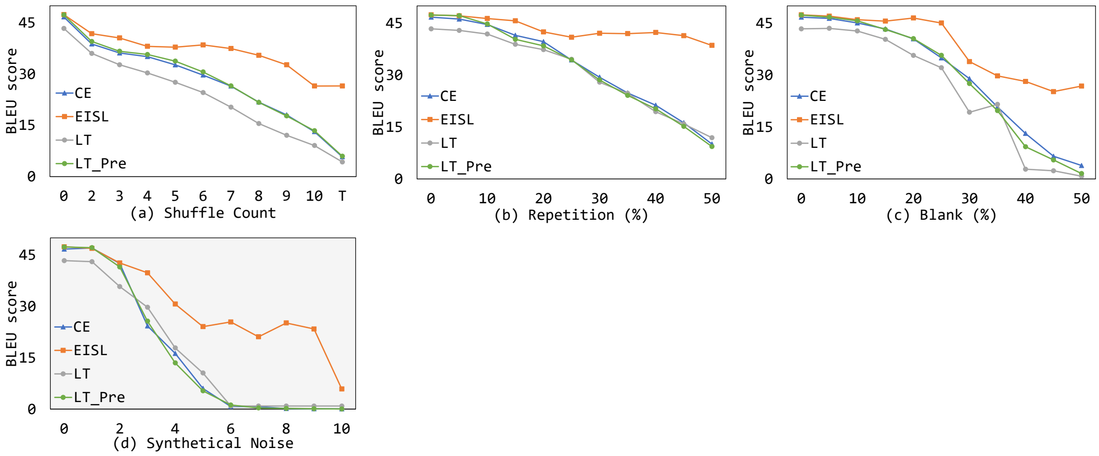

Figure 3.8: Comparison results with Loss Truncation(LT) of Translation with Noisy Target on German-to-English(de-en) from Multi30k. BLEU scores are computed against clean test data.

We can see Loss Truncation can sometimes slightly improve over CE, especially when the data is clean or with low/moderate noise. However, by simply ignoring high-loss data, LT is not good at handling data with high noise (which often leads to high loss). In comparison, our proposed EISL achieves a substantial improvement in the presence of high noise.

Reasons of Better Performance with Lower-gram EISL

In this section, we discuss the reason of why the performance of using lower grams is better than higher-gram EISL in Figure 3.5(e).

Lower-gram EISL is less sensitive to noise. For example, 1-gram EISL focuses mostly on matching individual tokens without caring much about the order of tokens; while a high-gram EISL (e.g., consider the extreme case of T^*-gram where T^* is the target length) reduces to CE (as discussed in Sec 3.2.3) and is highly sensitive to noise.

Thus, in the presence of high data noise, lower-gram EISL would be more robust and perform better.

Besides, on low-noise data (e.g., noise-level = 0 or 1), lower-gram EISL performs comparably with higher-gram EISL, both close to the CE performance. This is because we pretrained the model with CE (as mentioned in the experimental setup), and finetuning with EISL (either with lower- or higher-grams) would not change the performance a lot given the low-noise data.

Cases Study

As shown in Table 3.8, 3.9, 3.10, 3.11 and 3.12, we randomly sample some examples from generated sentences of the models trained with different types of noise on Multi30k dataset. For the sake of convenience, we use abbreviations in the tables, i.e., SC, RR, BR and NL are short for Shuffle Count, Repetition Ratio, Blank Ratio and Noise Level (for Synthetical Noise), respectively.

Shuffle Noise When there exist a few shuffle noises, e.g., SC $=$ 3, CE loss may lead word reduplicated (Example 1 and Example 2) and slightly wrong word order (Example 4 and Example 5), and there are some information mistranslated (*beautiful* in Example 4) or extra irrelevant information added (*black* in Example 5). As shuffle count increases, the aforementioned problems are increasingly severe, resulting the generated sentences meaningless. Especially, there are some words untranslated in PG examples (*eingezäunten* in Example 1, *irgendwo* in Example 2, *haben* in Example 5,). But EISL loss could keep the content consistency and grammatical correctness as far as possible.

Repetition Noise The main problem of the models trained by CE and PG with repetition noises is that the models can't filter the repetition noise out in training samples, and try to learn the wrong distribution, leading to generate reduplicated words frequently (Example 1-5). Specifically, the examples of CE and PG in RR $= 50\%$ are very representative. However, it's amazing that EISL can almost avoid such a problem even the repetition ratio achieves 50%. Meanwhile, the main semantics is preserved and the grammar is correct.

Blank Noise When adding blank noise, some tokens in targets will be substituted as *unk* so the targets will lose some information. We could measure from two aspects: one is the term frequency of meaningless token *unk* in generated sentences, and the other is the meaningful contents preserved by the models. Obviously, EISL loss handles better than CE loss on both aspects. Especially, when BR $= 20\%$, unlike models with CE, models with PG and EISL barely generate the *unk* token, and could translate the core content (Example 1-5). As BR increases, EISL could preserve more key information and produce less *unk* than CE and PG. Moreover, PG performs rather poor when BR is high (like BR $= 45\%$), and it almost loses all information (Example 1-5) and generates some confusing words (*teil* in Example 1, *afroamerikanischer* and *irgendwo* in Example 3, *beachaufsichtgebäude* in Example 4, and *holzstück* in Example 5).

Synthetical Noise We then evaluate the results of models trained by synthetical noise. Such a situation combines aforementioned three types of noises. One most highlighted advantage of EISL is that the generated sentences are almost grammatically correct and include main content as far as possible. However, CE can only stiffly joint some words, and can't guarantee the grammatical correctness (word order, word repetition and so on). PG performs worst, involving all the problems in CE cases and the meaningless word generation problem (Example 1-5).

3.4.4 Additional Results of Text Style Transfer

Examples on Yelp dataset

Some examples of generated sentences are given in Table 3.4. The model with EISL can select more appropriate adjective and improve the quality of the sentences. In the first example, the model should transfer the negative adjectives *cold* and *watery* to some positive adjectives that describe food. Obviously, the *delicious* is more appropriate than *excellent*. In the second example, the base model reverses both *not* and *stop*, leading to wrong sentiment and inconsistent content. While the model with EISL could avoid such a situation and generate more suitable sentence.

Source	my “ hot ” sub was *cold* and the meat was *watery* .
Base Model	my “ hot ” sub was *excellent* and the meat was *excellent* .
with EISL	my “ hot ” sub was *delicious* and the meat was *delicious* .
Source	the man did *not stop* her .
Base Model	the man did *definitely right* her .
with EISL	the man did *definitely stop* her .

Table 3.4: Examples of the generated sentences.

Results on Political dataset

Since the instances from democratic data and republican data are quite different, names of politicians have high correlation with the political slant. Therefore the BLEU score and POS distance have a big gap with the sentiment results. The results are shown in Table 3.5.

Model	Accuracy(%)	BLEU	PPL	POS distance
Prabhumoye et al. [2018]	86.5	7.38	-	7.298
Hu et al. [2017a]	**90.7**	47.50	-	3.524
Tian et al. [2018]	88.0	59.63	28.46	2.348
with EISL	89.2	**60.26**	**27.85**	**2.191**

Table 3.5: The results on the political dataset. The first two results are reported by [Tian et al., 2018].

3.4.5 Additional Results of Non-Autoregressive Generation

Results of Iterative NAT Models

As shown in Figure 3.9, with the increasing of iteration steps, the difference fades away.

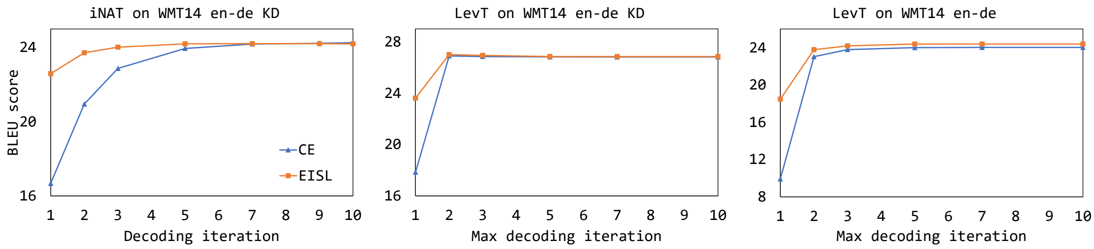

Figure 3.9: Results of iterative NAT on different decoding iterations.

Results of BLEURT Metric

To show the superiority of our method, We also evaluate on recent text generation metric, BLEURT [Sellam et al., 2020]. BLEURT is an evaluation metric for Natural Language Generation. It takes a pair of sentences as input, a reference and a candidate, and it returns a score that indicates to what extent the candidate is fluent and conveys the mearning of the reference. We use the recommended BLEURT-20 checkpoint. It gives a score for every sentence pair, and we averaged the scores to get the final score. The results are shown in Table 3.6.

Model	WMT14 en-de KD		WMT14 en-de	
	CE	EISL	CE	EISL
Vanilla-NAT [Gu et al., 2018]	0.346	**0.416**	0.194	**0.277**
NAT-CRF [Sun et al., 2019]	0.441	**0.464**	-	-
iNAT [Lee et al., 2018]	0.332	**0.437**	-	-
LevT [Gu et al., 2019]	0.355	**0.458**	0.214	**0.333**
CMLM [Ghazvininejad et al., 2019]	0.345	**0.450**	-	-

Table 3.6: The results (test set BLEURT) of EISL loss and CE loss applied to non-autoregressive models.

Qualitative Analysis on NAT Experiments

Given the non-autoregressive nature (i.e., all tokens are generated simultaneously), the one-to-one matching of CE loss can lead to severe mismatching. We consider the example: the predicted sentence is `a cat is on the red blanket` and the target sentence is `a cat is sitting on the red blanket`. The "on the red blanket" part of the prediction will be corrected to match the target positions, and this may lead to overcorrection (e.g., "on the red red blanket ."). Repetition is often a sign of overcorrection. However, with EISL, this situation will not happen because the phrase will be matched to appropriate target tokens. Let's have a look at a real example in Figure 3.10.

Source	Anja Schlichter managed the tournament
Target	Anja Schlichter leitet das Turnier
CE	Anja Schlichter leitdas Turnier Turnier
EISL	Anja Schlichter leitete das Turnier geleitet

Figure 3.10: Examples of the generated sentences.

Take the non-autoregressive model CMLM [Ghazvininejad et al., 2019] for example, we evaluate the translation of CMLM models trained by CE and EISL. As shown in Figure 3.11, our proposed EISL can reduce repetition to a large extent.

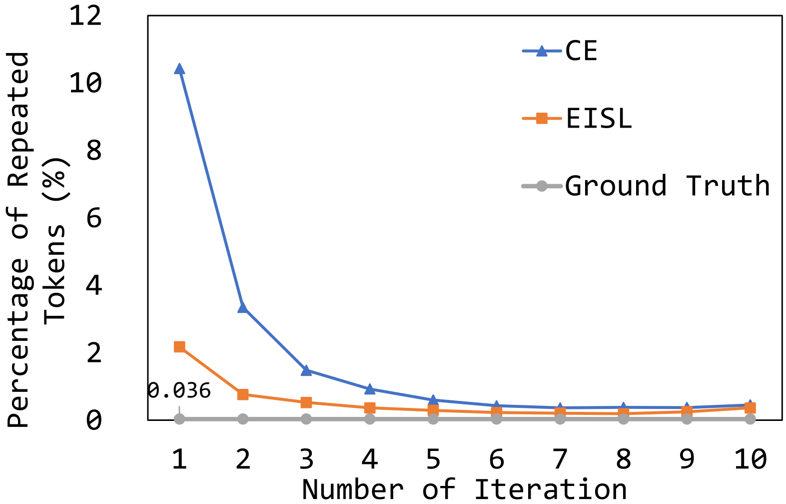

Figure 3.11: The percentage of repeated tokens under different iteration steps.

3.4.6 Efficiency Analysis

Complexity analysis Given T^* tokens, the time complexity of CE loss is $\mathbf{O}(T^*)$, while the complexity of n-gram EISL loss is $\mathbf{O}(n(T^* - n + 1)^2) \approx \mathbf{O}(T^{*2})$, assuming small n is used in practice (e.g., $n \in \{1, 2, 3, 4\}$). However, in practice, the computation cost of the loss (either CE or EISL) is **negligible** compared to the cost of model forward and backward during training. Thus, the extra cost introduced by the EISL loss is rather minor.

Empirical comparison of time cost To quantify the computational cost of different methods, we adopt CE and EISL on top of the same model and setting, and evaluate the consumed time for 1 training epoch. For comparison on both small and large dataset, we evaluate on Multi30k (29k training data, 1k test data) and 1M scale WMT-18 raw corpus (1M training data, 3k test data). The models are tested on one

Tesla V100 DGXS with 32 GB memory, the batch size is 128, max number of tokens is 6000 and update frequency is 4. For each method, we test 6 times and average the results as final time. The results are shown in Figure 3.12.

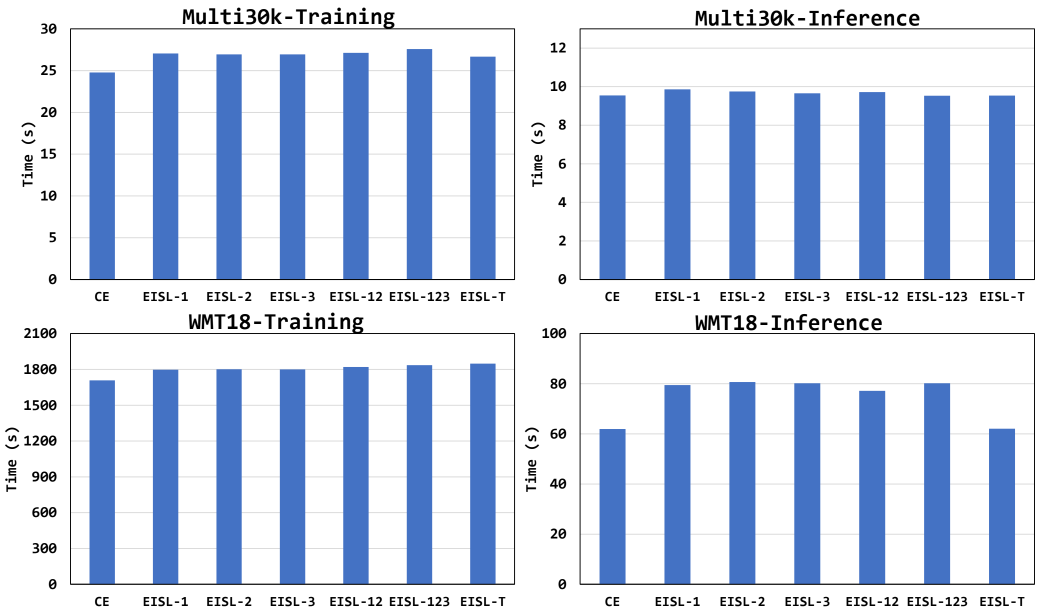

Figure 3.12: Results of training and inference time. EISL-n represents n-gram EISL loss and EISL-12 represents the combination of 1-gram and 2-gram EISL loss.

Empirical total time cost of EISL training As discussed in the experiments in the paper, we first pretrain the model with the CE loss until convergence, and then finetune with the EISL loss. Here we report the total time cost of each stage, based on the WMT-18 translation setting as described in Section 3.3.1. The results are shown in Table 3.7. As the data size increases, the convergence time of both pretraining and finetuning grows. The time cost of the finetuning stage is less than half of that of the pretraining stage.

Data Size	PreTraining Time (CE)	Finetuning Time (EISL)
1M	1h 40min 57s	49min 33s
2M	5h 56min 57s	1h 35min 10s
4M	8h 55min 18s	3h 57min 44s

Table 3.7: Convergence time of pretraining and finetuning stages.

3.4.7 Hyperparameters

Regarding which n-grams to use and their weights w_n in the EISL loss, we found in our experiments that the default values *largely* following the standard BLEU metric (i.e., maximum $n = 4$ with equal weights) work well. Specifically, we use $n \in \{2, 3, 4\}$ and equal weights $w_n = 1/3$ as our default values. Most of our experiments adopt the default values which achieve consistent substantial improvement over CE and other rich baselines as shown in our experiments. (except for the synthetic experiment where we show the effect of different n-grams including those selected using the validation set).

Besides, in our experiments, we first pretrain the model with the CE loss (i.e., EISL with $n = T^*$ and teacher forcing, see Section 3.2.3) and then finetune with the EISL loss. We simply do the CE pretraining *until convergence* before switching to the EISL finetuning. Therefore, there is no need of tuning for the training iterations of pretraining.

3.4.8 Analysis of Efficient Implementation

In order to validate the efficiency and accuracy of our approximation (for autoregressive models) discussed in Section 3.2.2, we conduct the analysis experiments, showing that the approximate (and efficient) EISL loss values are very close to exact (but expensive) EISL value. We use the same setting as section 3.3.1, and finetune the model with our efficient approximate EISL loss on Multi30k. Throughout the course of training, we record the loss values of both the exact implementation and our approximate implementation. As shown in Figure 3.13(a) and (b), the tendency of two losses is very close to each other. We also plot the absolute difference of the two losses as shown in Figure 3.13(c). We can see the difference decreases as training proceeds. The observations validate the effectiveness of our approximate implementation.

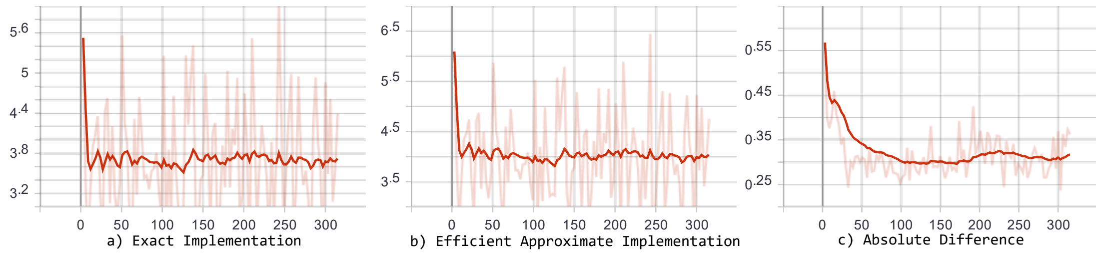

Figure 3.13: The change of loss values during training. The x-axis represents the training step. a) gives the loss curve of exact implementation; b) gives the loss curve of efficient approximate implementation as we discussed in section 3.2.2; and c) gives the absolute difference between the two implementations.

We note that training the model with the exact loss is costly, which necessitates our approximation. Specifically, for n-gram loss, we need to run the forward pass of the decoder $(T-n)^2$ times, and keep the whole computation graph for back-propagation, which will consume much more time and memory. Even for only loss evaluation (without the backward pass), we found the runtime of the exact loss is about 15 times longer than that of the efficient approximate implementation based on convolution operator.

3.5 Conclusions

In this chapter, we have introduced our new approach, Edit-Invariant Sequence Loss (EISL), for end-to-end training of neural text generation models. Our proposed method is designed to be insensitive to the shift of n-grams in target sequences, making it suitable for training with noisy data and weak supervisions where CE loss often fails. We have shown that CE loss is a special case of EISL and established a connection between EISL with BLEU metric and convolution operation, both of which have the invariant property. Through experiments on translation with noisy targets, text style transfer, and non-autoregressive neural machine translation, we have demonstrated the superiority of our method. Our work offers promising results, but more general applications and further exploration of the superiority of EISL on other diverse text generation problems remain to be studied, such as compositional generalization [Andreas et al., 2019] and causal invariance [Hu and Li, 2021] in language. These fundamental challenges present exciting avenues for future research in this area.

□ **End of chapter.**

Source (de)		ein junger mann nimmt an einem lauf teil und derjenige , der dies aufzeichnet , lächelt .
Target (en)		**a young man participates in a career while the subject who records it smiles .**
SC = 3	CE	young man is running on a a and the other man is smiling .
	PG	young man is running on a track and the other man is smiling .
	EISL	young man is running in a dirt course and the other is smiling .
SC = 6	CE	young man is running a a race and the other is smiling .
	PG	young man taking a race and the other smiling . a
	EISL	young man is running a race and the other guy is smiling .
SC = 9	CE	young man . a a the is running up and up hill smiling taking
	PG	young man takes on a slope and thejenige , the the smiles . a
	EISL	young man is on a hillside smiling and the others , who is smiling .
RR = 15%	CE	young man is running on a track and the other is smiling .
	PG	young man is running on a track and the other is smiling .
	EISL	young man is running in a race and the runner is smiling .
RR = 30%	CE	young man man is is running on a track track and the the other is is smiling smiling .
	PG	young man man is is running on a track track and the other man man who is is is smiling .
	EISL	young man is running in a race and the other is smiling at him . .
RR = 50%	CE	a young young man man is is smiling smiling at at a a window window while another smiles smiles at him him . .
	PG	a young man man is is napping napping on on a a grassy grassy field field and and some people people are are smiling smiling . .
	EISL	young man running in a race and the other is smiling at the action . .
BR = 20%	CE	young man unk unk a run and the unk is smiling .
	PG	young man is running in a race and the one who is looking at him is smiling .
	EISL	young man is running in a race with the runner who is up .
BR = 35%	CE	young man unk unk a unk , and the unk is smiling unk
	PG	young man unk unk track unk others unk .
	EISL	young man unk is un in a race and the other un is un at the finish .
BR = 45%	CE	young unk is unk on a unk unk and the unk smiles unk
	PG	young man unk a unk teil unk unk .
	EISL	young unk un is un in a race , the other is smiling back .
NL = 5	CE	young man is running a race and the one who is running is smiling .
	PG	young man is running a race and the one scoring is smiling .
	EISL	young man is running a race and one of the runners is up to him .
NL = 15	CE	young man is unk unk a unk and the other man is smiling .
	PG	young man is on a unk smiling at thejenige . .
	EISL	young man is in a race , the other smiling .
NL = 20	CE	a young man is unk unk a unk and unk is smiling at him .
	PG	young smiles on in ail and thejenige smile on . . .
	EISL	young man unk unk a ladder and unk , who is unk smiling .

Table 3.8: Example 1.

Source (de)		15 große hunde spielen auf einem eingezäunten grundstück neben einem haus .
Target (en)		**15 large dogs playing in a fenced yard beside a house .**
SC = 3	CE	large dogs play on a a dirt path next to a house .
	PG	15 large dogs play on an earthen platform next to a house .
	EISL	large dogs are playing on a dirt path next to a house .
SC = 6	CE	large dogs play on a a play area next to abandoned house .
	PG	15 large dogs playing on a eingezäunten group stage next to a house .
	EISL	group of dogs play on a abandoned path next to a house .
SC = 9	CE	large dogs play a . on a field next to abandoned house
	PG	dogs play on a snowy grundstück next to a house .15 large
	EISL	. 15 large dogs play on an abandoned hillside next to a house .
RR = 15%	CE	large dogs are playing on a fenced in area next to a house .
	PG	large dogs are playing on a fenced in area next to a house .
	EISL	large dogs are playing on a fenced track next to a house .
RR = 30%	CE	large dogs dogs play on on a a dirt track near a house house .
	PG	large dogs dogs play on a fenced-in area area next to a house .
	EISL	large dogs play on a fenced walkway next to a house . .
RR = 50%	CE	small dogs dogs play on on a a grassy grassy field field next next to to a house house . .
	PG	15 large dogs dogs are are playing playing on on a a grassy grassy field field next next to to a house house . .
	EISL	15 large dogs playing on a fenced terrain next to a house . .
BR = 20%	CE	large dogs play in a fenced yard next to a house .
	PG	large dogs are playing on an overcast walk next to a house .
	EISL	large dogs are playing in a fenced area near to a house .
BR = 35%	CE	unk dogs play unk a unk unk by a house .
	PG	large dogs unk a unk path unk unk house .
	EISL	large dogs unk play in a fenced area next to a house .
BR = 45%	CE	unk dogs unk on a unk unk next to unk house .
	PG	large dogs unk a unk unk .
	EISL	large unk un are un in a fenced-out game next to a house .
NL = 5	CE	large dogs are playing on a fenced in area next to a house .
	PG	large dogs are playing on a fenced in area next to a house .
	EISL	large dogs are playing on a fenced backwalk next to a house .
NL = 15	CE	large dogs are playing on a unk grassy field next to a house .
	PG	large dogs playing on a unk next to a house . . .
	EISL	large dogs play on a covered piece of furniture next to a house .
NL = 20	CE	large dogs are playing on on a a a grassy grassy field next to a house .
	PG	large play play in auntenck in a house . . .
	EISL	large dogs play on a unk unk next to a house . .

Table 3.9: Example 2.

Source (de)		ein afroamerikanischer mann spielt irgendwo in der stadt gitarre und singt
Target (en)		**an african american man playing guitar and singing in an urban setting .**
SC = 3	CE	african american man is playing the guitar and singing in the city .
	PG	african american man is playing the guitar in the city and singing
	EISL	african american man is playing the guitar in the city and singing .
SC = 6	CE	african-american man is playing guitar in the a and singing city .
	PG	african american man playing irgendwo in the city guitar singing
	EISL	african american man is playing the guitar in the city
SC = 9	CE	african-american man playing guitar in the a and singing city
	PG	african americanischer man plays irgendwo in the city guitar singing . a
	EISL	african american man is playing the guitar in the city and singing
RR = 15%	CE	african american american man plays guitar guitar in the city city .
	PG	african american man is playing guitar in the city and singing .
	EISL	african american man is playing guitar in the city and singing .
RR = 30%	CE	african american man plays guitar guitar in in the city city while singing .
	PG	african american man man plays guitar guitar in the city city and sings .
	EISL	an african american man playing guitar in the city and singing . .
RR = 50%	CE	african african american american man playing guitar guitar in in the the city city and singing singing .
	PG	african american american man man is is playing playing guitar guitar in in the the city city . .
	EISL	an african american man playing guitar in the city and singing . .
BR = 20%	CE	african american man plays guitar unk sings unk
	PG	african american man is playing guitar and singing in the city .
	EISL	african american man is playing the guitar and singing .
BR = 35%	CE	african american man unk unk guitar unk singing unk
	PG	african american man unk guitar unk singing unk
	EISL	african american unk is un a guitar and singing in the city .
BR = 45%	CE	african american unk unk playing unk guitar in unk city unk
	PG	afroamerikanischer man unk irgendwo unk unk
	EISL	af unk un playing some sort of guitar in the city and singing .
NL = 5	CE	african american man plays guitar and sings somewhere in the city .
	PG	african american man is playing guitar and singing in the city .
	EISL	african american man is playing guitar and singing somewhere in the city .
NL = 15	CE	african american man is playing the guitar in the city and singing .
	PG	afroamerikanischer man is irgendwo in the city guitarre .
	EISL	african american man playing some sort of guitar in the city and singing .
NL = 20	CE	african american american man is playing the guitar in the the city unk
	PG	afroamerikanischer singt in the city guitarre singt .
	EISL	african american man plays unk unk in the city unk

Table 3.10: Example 3.

Source (de)		ein strandaufsichtgebäude steht im sand , es ist ein bewölkter tag .
Target (en)		**a lifeguard building is on the sand on a cloudy day .**
SC = 3	CE	beach a is standing in the sand on a beautiful day .
	PG	beachfront building is standing in the sand on a beautiful day .
	EISL	beach view building is standing in the sand on a cloudy day .
SC = 6	CE	beach a is in the sand building on a beautiful day .
	PG	beach viewgeb building standing in sand on a beautiful day .
	EISL	beach view building is standing in the sand on a beautiful day .
SC = 9	CE	beach a in the sand . a cloudy day stands beach
	PG	beachaufsichtge building stands in sand , the is a beautiful day . a
	EISL	. a beachfront building standing in the sand is a beautiful day .
RR = 15%	CE	beachfront building is standing in the sand on a cloudy day .
	PG	beachfront building is standing in sand , it is a cloudy day .
	EISL	beach building is standing in the sand , it is a cloudy day .
RR = 30%	CE	beachfront beachfront building building is is standing standing in the sand sand on a cloudy day .
	PG	beachfront beachfront building building is standing in sand sand on a cloudy day .
	EISL	beachfront building is standing in the sand , it is a cloudy day . .
RR = 50%	CE	a beachfront beachfront building building is is standing standing in in the sand sand , it looks like it is is a beach resort resort . .
	PG	a beachfront beachfront building building is is standing standing in in sand sand . .
	EISL	a beach view building is in the sand , it is a cloudy day . .
BR = 20%	CE	beachfront building is standing in sand on a cloudy day unk
	PG	beachfront building is standing in sand on a cloudy day .
	EISL	beach view building is standing in the sand , it is a cloudy day .
BR = 35%	CE	beach unk unk standing in sand on a cloudy day unk
	PG	beach unk building unk unk sand unk a cloudy day .
	EISL	beach building unk is un in the sand on a cloudy day .
BR = 45%	CE	unk unk is standing unk the sand unk it is a beautiful day unk
	PG	beachaufsichtgebäude unk unk sand unk .
	EISL	beach unk un is un in the sand , this is a cloudy day .
NL = 5	CE	beachfront view building is standing in the sand on a cloudy day .
	PG	beachfront view building is standing in sand on a cloudy day .
	EISL	beachfront building is standing in the sand , it is a cloudy day .
NL = 15	CE	beach unk unk is standing in the sand unk it is a sunny day .
	PG	beach unk is in sand on a snowy day . .
	EISL	beach building is in the sand , it is a cloudy day .
NL = 20	CE	beach unk unk is standing in the sand unk it is a sunny sunny day .
	PG	beachaufsichtgebäude steht in sand , es is a day . .
	EISL	beach unk stands in sand unk it is a sunny day . .

Table 3.11: Example 4.

Source (de)		zwei hunde haben beim spielen dasselbe holzstück im maul .
Target (en)		**two dog is playing with a same chump on their mouth .**
SC = 3	CE	dogs are two playing with . pieces of wood in their mouths two
	PG	dogs are playing with pieces of black wood in their mouths .
	EISL	two dogs are playing with pieces of wood in their mouths .
SC = 6	CE	dogs are two . playing with sticks in their mouths two
	PG	dogs have been playing with pieces of wood in their mouths . two
	EISL	two dogs are playing with pieces of wood in their mouths .
SC = 9	CE	two dogs their . are playing with sticks in muzzled
	PG	dogs haben beim play pieces in their mouth . two
	EISL	. two dogs have been playing with sticks in their mouth .
RR = 15%	CE	two dogs are are playing with a a piece piece of wood in their mouth .
	PG	dogs are playing with white wooden blocks in their mouth .
	EISL	two dogs are playing with some pieces of wood in their mouths .
RR = 30%	CE	two dogs dogs are are playing with a a piece piece of of wood in their mouths .
	PG	dogs dogs are are playing with white wooden blocks blocks in their mouth .
	EISL	two dogs are playing with pieces of wood in their mouths . .
RR = 50%	CE	two dogs dogs are are playing playing with with plastic plastic sticks sticks in in their their mouth mouth . .
	PG	two dogs dogs are are playing playing with with plastic holsters holsters in in their maul maul . .
	EISL	two dogs have playing with some white wood in their mouths . .
BR = 20%	CE	dogs unk unk pieces of wood in their mouths .
	PG	dogs are playing with wet wood in their mouths .
	EISL	dogs are playing with wet pieces of wood in their mouths .
BR = 35%	CE	unk have unk pieces of unk in their mouths .
	PG	two dogs unk unk piece of wood unk their mouth .
	EISL	two dogs unk playing with some piece of wood in their mouth .
BR = 45%	CE	dogs are playing with unk unk in unk mouth unk
	PG	dogs unk unk piece of unk holzstück unk .
	EISL	dogs unk un are un while play with some wood pieces in their mouth .
NL = 5	CE	two dogs are playing with the same piece of wood in their mouths .
	PG	dogs have pieces of of wood in their mouths .
	EISL	two dogs are playing with the same piece of wood in their mouths .
NL = 15	CE	two dogs are are are playing with unk unk in their mouths .
	PG	dogs haben on a game unk unk . . .
	EISL	two dogs have been playing with a piece of wood in their mouth .
NL = 20	CE	two dogs are are are playing with unk unk in their mouths .
	PG	dogs haben in a playenselbeck in their mouth . .
	EISL	two dogs are playing with unk sticks in their mouths . .

Table 3.12: Example 5.

Chapter 4

Composable Text Controls in Latent Space with ODEs

4.1 Introduction

Many text problems involve a diverse set of text control operations, such as editing different attributes (e.g., sentiment, formality) of the text, inserting or changing the keywords, generating new text of diverse properties, and so forth. In particular, different *composition* of those operations are often required in various real-world applications (Figure 4.1).

Conventional approaches typically build a conditional model (e.g., by finetuning pretrained language models) for each specific combination of operations [Hu et al., 2017b; Keskar et al., 2019; Ziegler et al., 2019], which is unscalable given the combinatorially many possible compositions and the lack of supervised data. Most recent research thus has started to explore plug-and-play solutions. Given a pretrained language model (LM), those approaches plug in arbitrary constraints to guide the production of desired text sequences [Dathathri et al., 2020; Yang and Klein, 2021; Kumar et al., 2021; Krause et al., 2021; Mireshghallah et al., 2022; Qin et al., 2022]. The

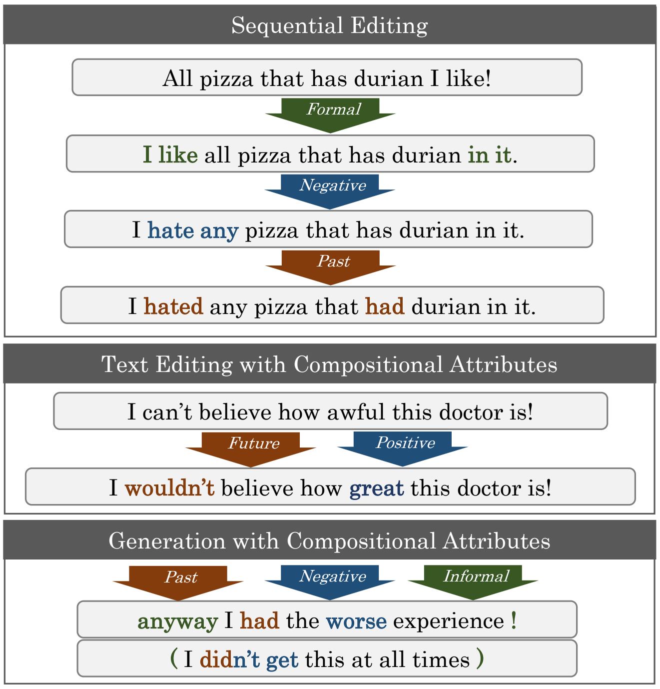

Figure 4.1: Examples of different composition of text operations, such as editing a text in terms of different attributes sequentially (top) or at the same time (middle), or generating a new text of target properties (bottom). The proposed LatentOps enables a single LM (e.g., an adapted GPT-2) to perform arbitrary text operation composition in the latent space.

approaches, however, typically rely on search or optimization in the complex text *sequence space*. The discrete nature of text makes the search/optimization extremely difficult. Though some recent work introduces continuous approximations to the discrete tokens [Qin et al., 2020, 2022; Kumar et al., 2021], the high dimensionality and complexity of the sequence space still renders it inefficient to find the accurate high-quality text.

In this work, we develop LatentOps, a new efficient approach that performs composable control operations in the compact and continuous *latent space* of text. LatentOps permits plugging in arbitrary operators (e.g., attribute classifiers) applied on text latent vectors, to form an energy-based distribution on the low-dimensional latent space. We then develop an efficient sampler based on ordinary differential equations (ODEs) [Song et al., 2021; Nie et al., 2021; Vahdat et al., 2021] to draw latent vector samples that bear the desired attributes.

A key challenge after getting the latent vector is to decode it into the target text sequence. To this end, we connect the latent space to pretrained LM decoders (e.g., GPT-2) by efficiently adapting a small subset of the LM parameters in a variational auto-encoding (VAE) manner [Kingma and Welling, 2014; Bowman et al., 2016].

Previous attempts of editing text in latent space have often been limited to single attribute and small-scale models, due to the incompatibility of the latent space with the existing transformer-based pretrained LMs [Wang et al., 2019a; Liu et al., 2020; Shen et al., 2020; Duan et al., 2020; Mai et al., 2020b]. LatentOps overcomes the difficulties and enables a single large LM to perform arbitrary composable text controls.

We conduct experiments on three challenging settings, including sequential editing of text *w.r.t.* a series of attributes, editing compositional attributes simultaneously, and generating new text given various attributes. Results show that composing operators within our method manages to generate or edit high-quality text, substantially improving over respective baselines in terms of quality and efficiency.

4.2 Technical Background

4.2.1 Energy-based Models and ODE Sampling

Given an arbitrary energy function $E(\boldsymbol{x}) \in \mathbb{R}$, energy-based models (EBMs) define a Boltzmann distribution:

$$p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})}/Z, \tag{4.1}$$

where $Z = \sum_{\boldsymbol{x} \in \mathcal{X}} e^{-E(\boldsymbol{x})}$ is the normalization term (the summation is replaced by integration if $\boldsymbol{x} \in \mathcal{X}$ is a continuous variable). EBMs are flexible to incorporate any functions or constraints into the energy function $E(\boldsymbol{x})$. Recent work has explored text-based EBMs (where $\boldsymbol{x}$ is a text sequence) for controllable text generation [Hu et al., 2018; Deng et al., 2020; Khalifa et al., 2021; Mireshghallah et al., 2022; Qin et al., 2022]. Despite the flexibility, sampling from EBMs is rather challenging due to the intractable Z. The text-based EBMs face with even more difficult sampling due to the extremely large and complex (discrete or soft) text space.

Langevin dynamics [LD, Welling and Teh, 2011; Ma et al., 2018] is a gradient-based Markov chain Monte Carlo (MCMC) approach often used for sampling from EBMs [Du and Mordatch, 2019b; Song and Ermon, 2019; Du et al., 2020; Qin et al., 2022]. It is considered as a more efficient way compared to other gradient-free alternatives (e.g., Gibbs sampling [Bishop and Nasrabadi, 2006]). However, due to several critical hyperparameters (e.g., step size, number of steps, noise scale), LD tends to be sensitive and unrobust in practice [Nie et al., 2021; Du and Mordatch, 2019a; Grathwohl et al., 2020].

On the other hand, stochastic/ordinary differential equations (SDEs/ODEs) [Anderson, 1982] offer another sampling technique recently applied in image generation [Song et al., 2021; Nie et al., 2021]. An SDE characterizes a *diffusion process* that maps real data to random noise in continuous time $t \in [0, T]$. Specifically, let $\boldsymbol{x}(t)$ be the value of the process following $\boldsymbol{x}(t) \sim p_t(\boldsymbol{x})$, indexed by time t. At start time $t = 0$, $\boldsymbol{x}(0) \sim p_0(\boldsymbol{x})$ which is the data distribution, and at the end $t = T$, $\boldsymbol{x}(T) \sim p_T(\boldsymbol{x})$ which is the noise distribution (e.g., standard Gaussian). The *reverse* SDE instead generates a real sample from the noise by working backwards in time (from $t = T$ to $t = 0$). More formally, consider a *variance-preserving* SDE [Song et al., 2021] whose reverse is written as

$$\mathrm{d}\boldsymbol{x} = -\frac{1}{2}\beta(t)[\boldsymbol{x} + 2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{\boldsymbol{w}}, \tag{4.2}$$

where $\mathrm{d}t$ is an infinitesimal negative time step; $\bar{\boldsymbol{w}}$ is a standard Wiener process when time flows backwards from T to 0; and the scalar $\beta(t) := \beta_0 + (\beta_T - \beta_0)t$ is a time-variant coefficient linear *w.r.t.* time t. Given a noise $\boldsymbol{x}(T) \sim p_T(\boldsymbol{x})$, solving the above reverse SDE returns a $\boldsymbol{x}(0)$ that is a sample from the desired distribution $p_0(\boldsymbol{x})$. One could use different numerical solvers to this end. [Burrage et al., 2000; Higham, 2001; Rößler, 2009]. The SDE sampler sometimes need to combine with an additional corrector to improve the sample quality [Song et al., 2021].

Further, as shown in [Song et al., 2021; Maoutsa et al., 2020], each (reverse) SDE has a corresponding ODE, solving which leads to samples following the same distribution. The ODE is written as (see Appendix 4.5.1 for the derivations):

$$\mathrm{d}\boldsymbol{x} = -\frac{1}{2}\beta(t)[\boldsymbol{x} + \nabla_x \log p_t(\boldsymbol{x})]\mathrm{d}t. \tag{4.3}$$

Solving the ODE with relevant numerical methods [Euler, 1824; Calvo et al., 1990; Engstler and Lubich, 1997] corresponds to an sampling approach that is more efficient and robust [Song et al., 2021; Nie et al., 2021].

In this work, we adapt the ODE sampling for our approach. Crucially, we overcome the text control and sampling difficulties in the aforementioned sequence-space methods, by defining the text control operations in a compact latent space, handled by a latent-space EBMs with the ODE solver for efficient sampling.

4.2.2 Latent Text Modeling with Variational Auto-Encoders

Variational auto-encoders (VAEs) [Kingma and Welling, 2014; Rezende et al., 2014] have been used to model text with a low-dimensional continuous latent space with certain regularities [Bowman et al., 2016; Hu et al., 2017b]. An VAE connects the text sequence space $\mathcal{X}$ and the latent space $\mathcal{Z} \subset \mathbb{R}^d$ with an encoder $q(\boldsymbol{z}|\boldsymbol{x})$ that maps text $\boldsymbol{x}$ into latent vector $\boldsymbol{z}$, and a decoder $p(\boldsymbol{x}|\boldsymbol{z})$ that maps a $\boldsymbol{z}$ into text. Previous work usually learns text VAEs from scratch, optimizing the encoder and decoder parameters with the following objective:

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{x}) = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] + \text{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p_{\text{prior}}(\boldsymbol{z})), \tag{4.4}$$

where $p_{\text{prior}}(\boldsymbol{z})$ is a standard Gaussian distribution as the prior, and $\text{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence that pushes q_{enc} to be close to the prior. The first term encourages $\boldsymbol{z}$ to encode relevant information for reconstructing the observed text $\boldsymbol{x}$, while the second term adds regularity so that any $\boldsymbol{z} \sim p_{\text{prior}}(\boldsymbol{z})$ can be decoded into high-quality text in the text sequence space $\mathcal{X}$. Recent work [Li et al., 2020; Hu and Li, 2021] scales up VAE by initializing the encoder and decoder with pretrained LMs (e.g., BERT [Devlin et al., 2019] and GPT-2 [Radford et al., 2019b], respectively). However, they still require costly finetuning of the whole model on the target corpus.

In comparison, our work converts a given pretrained LM (e.g., GPT-2) into a latent-space model efficiently by tuning only a small subset of parameters, as detailed more in §4.3.3.

4.3 Composable Text Latent Operations

We develop our approach LatentOps that quickly adapts a given pretrained LM (e.g., GPT-2) to enable composable text latent operations. The approach consists of two components, namely a VAE based on the pretrained LM that connects the text space with a compact continuous latent space, and EBMs on the latent space that permits arbitrary attribute composition and efficient sampling.

More specifically, the VAE decoder $p(\boldsymbol{x}|\boldsymbol{z})$ offers a way to map any given latent vector $\boldsymbol{z}$ into the corresponding text sequence. Therefore, text control (e.g., editing a text or generating a new one) boils down to finding the desired vector $\boldsymbol{z}$ that bears the desired attributes and characteristics. To this end, one could plug in any relevant attribute operators (e.g., classifiers), resulting in a latent-space EBM that characterizes the distribution of $\boldsymbol{z}$ with the desired attributes. We could then draw the $\boldsymbol{z}$ samples of interest, performed efficiently with an ODE solver. Figure 4.2 gives an illustration of the approach.

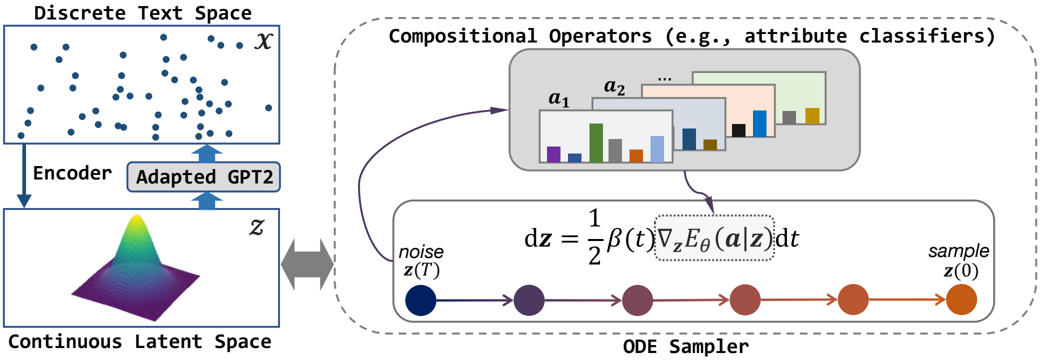

Figure 4.2: Overview of LatentOps. (Left): We equip pretrained LMs (e.g., GPT-2) with the compact continuous latent space through parameter-efficient adaptation (§4.3.3). (Right): One could plug in arbitrary operators (e.g., attribute classifiers) to obtain the latent-space EBM (§4.3.1). We then sample desired latent vectors efficiently by solving the ODE which works backwards through the diffusion process from time $t = T$ to 0. The resulting sample $\boldsymbol{z}(0)$ is fed to the decoder (adapted GPT-2) to generate the desired text sequence.

LatentOps thus avoids the difficult optimization or sampling in the complex text sequence space as compared to the previous plug-and-play methods [e.g., Yang and Klein, 2021; Dathathri et al., 2020; Qin et al., 2022]. Our approach is also compatible with the powerful pretrained LMs, requiring only minimal adaptation to equip the LMs with a latent space, rather than costly retraining from scratch as in the recent diffusion LM [Li et al., 2022].

In the following, we first present the latent-space EBM formulation (§4.3.1) for composable operations, and derive the efficient ODE sampler (§4.3.2); we discuss the parameter-efficient adaptation of pretrained LMs for the latent space (§4.3.3); we then discuss the implementation details (§4.3.4).

4.3.1 Composable Latent-Space EBMs

We aim to formulate the latent-space EBMs such that one can easily plug in arbitrary attribute operators to define the latent distribution of interest. Besides, as we want to obtain fluent text with the VAE decoder $p(\boldsymbol{x}|\boldsymbol{z})$ described in §4.3.3, the latent distribution over $\boldsymbol{z}$ should match the structure of the VAE latent space.

Formally, let $\boldsymbol{a} = \{a_1, a_2, ...\}$ be a vector of desired attribute values, where each $a_i \in \mathbb{R}$ (e.g., positive sentiment, or informal writing style). Note that $\boldsymbol{a}$ does not have a prefixed length as one can plug in any number of attributes to control on the fly. In general, to assess if a vector $\boldsymbol{z}$ bears the desired attribute a_i, we could use any function f_i that takes in $\boldsymbol{z}$ and a_i, and outputs a score measuring how well a_i is carried in $\boldsymbol{z}$. For a categorical attribute (e.g., sentiment, either positive or negative), one of the common choices is to use a trained attribute classifier, where $f_i(\boldsymbol{z})$ is the output logit vector and $f_i(\boldsymbol{z})[a_i] \in \mathbb{R}$ is the logit of the particular class a_i of interest. For clarity of presentation, we focus on categorical attributes and classifiers in the rest of the paper, and assume the attributes are independent with each others.

We are now ready to formulate the latent-space EBMs by plugging in the attribute classifiers. Specifically, we define the joint distribution:

$$p(\boldsymbol{z}, \boldsymbol{a}) := p_{\text{prior}}(\boldsymbol{z}) p(\boldsymbol{a}|\boldsymbol{z}) = p_{\text{prior}}(\boldsymbol{z}) \cdot e^{-E(\boldsymbol{a}|\boldsymbol{z})} / Z, \tag{4.5}$$

where $p_{\text{prior}}(\boldsymbol{z})$ is the Gaussian prior distribution of VAE (§4.2.2), and $p(\boldsymbol{a}|\boldsymbol{z})$ is formulated with energy function $E(\boldsymbol{a}|\boldsymbol{z})$ to encode the different target attributes. Such a decomposition of $p(\boldsymbol{z}, \boldsymbol{a})$ results in two key desirable properties: (1) The marginal distribution over $\boldsymbol{z}$ equals the VAE prior, i.e., $\sum_{\boldsymbol{a}} p(\boldsymbol{z}, \boldsymbol{a}) = p_{\text{prior}}(\boldsymbol{z})$. This facilitates the VAE decoder to generate fluent text; (2) the energy function in $p(\boldsymbol{a}|\boldsymbol{z})$ enables the combination of arbitrary attributes, with $E(\boldsymbol{a}|\boldsymbol{z}) = \sum_i \lambda_i E_i(a_i|\boldsymbol{z})$. Each $\lambda_i \in \mathbb{R}$ is the balance weight, and E_i is the defined as the negative log probability (i.e., the normalized logit) of a_i to make sure the different attribute classifiers have outputs at the same scale for combination:

$$E_i(a_i|\boldsymbol{z}) = -f_i(\boldsymbol{z})[a_i] + \log \sum\nolimits_{a_i'} \exp(f_i(\boldsymbol{z})[a_i']). \tag{4.6}$$

4.3.2 Efficient Sampling with ODEs

Once we have the desired distribution $p(\boldsymbol{z}, \boldsymbol{a})$ over the latent space and attributes, we would like to draw samples $\boldsymbol{z}$ given the target attribute values $\boldsymbol{a}$. The samples can then be fed to the VAE decoder (§4.3.3) to obtain the desired text. As discussed in §4.2.1 and also shown in our ablation study in §4.5.3, sampling with ODEs has the benefits of robustness compared to Langevin dynamics that is sensitive to hyperparameters, and efficiency compared to SDEs that require additional correction.

We now derive the ODE sampling in the latent space. Specifically, we adapt the ODE from Eq.(4.3) into our latent-space setting, which gives:

$$\begin{aligned} \mathrm{d}\boldsymbol{z} &= -\frac{1}{2}\beta(t)[\boldsymbol{z} + \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z}, \boldsymbol{a})]\mathrm{d}t \\ &= -\frac{1}{2}\beta(t)\left[\boldsymbol{z} + \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{a}|\boldsymbol{z}) + \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z})\right]\mathrm{d}t. \end{aligned} \tag{4.7}$$

For $p_t(\boldsymbol{z})$, notice that at $t = 0$, $p_0(\boldsymbol{z})$ is the VAE prior distribution $p_{\text{prior}}(\boldsymbol{z})$ as defined in Eq.(4.5), which is the same as $p_T(\boldsymbol{z})$ (i.e., the Gaussian noise distribution after diffusion). This means that in the diffusion process, we always have $p_t(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, I)$ that is time-invariant [Nie et al., 2021]. Similarly, for $p_t(\boldsymbol{a}|\boldsymbol{z})$, since the input $\boldsymbol{z}$ follows the time-invariant distribution and the classifiers f_i are fixed, the $p_t(\boldsymbol{a}|\boldsymbol{z})$ is also time-invariant. Plugging the definitions of those components, we obtain the simple ODE formulation:

$$\begin{aligned} \mathrm{d}\boldsymbol{z} &= -\frac{1}{2}\beta(t)[\boldsymbol{z} - \nabla_{\boldsymbol{z}} E(\boldsymbol{a}|\boldsymbol{z}) - \frac{1}{2}\nabla_{\boldsymbol{z}}||\boldsymbol{z}||_2^2]\mathrm{d}t \\ &= \frac{1}{2}\beta(t)\sum_{i=1}^{n} \nabla_{\boldsymbol{z}} E(a_i|\boldsymbol{z})\mathrm{d}t. \end{aligned} \tag{4.8}$$

We can then easily create latent samples conditioning on the given attribute values, by drawing $\boldsymbol{z}(T) \sim \mathcal{N}(\mathbf{0}, I)$ and solving the Eq.(4.8) with a differentiable neural ODE solver[1] [Chen et al., 2018, 2021] to obtain $\boldsymbol{z}(0)$. In §4.3.4, we discuss more implementation details with approximated starting point $\boldsymbol{z}(T)$ for text editing and better empirical performance.

[1] https://github.com/rtqichen/torchdiffeq

4.3.3 Adapting Pretrained LMs for Latent Space

To decode the $\boldsymbol{z}$ samples into text sequences, we equip pretrained LMs (e.g., GPT-2) with the latent space through parameter-efficient adaptation. More specifically, we adapt the autoregressive LM into a text latent model within the VAE framework (§4.2.2). Differing from the previous VAE work that trains from scratch or finetunes the full parameters of pretrained LMs [Li et al., 2020; Hu and Li, 2021; Hu et al., 2017b], we show that it is sufficient to only update a small portion of the LM parameters to connect the LM with the latent space, while keeping the LM capability of generating fluent coherent text. Specifically, we augment the autoregressive LM with small MLP layers that pass the latent vector $\boldsymbol{z}$ to the LM, and insert an additional transformer layer in between the LM embedding layer and the original first layer. The resulting model then serves as the decoder in the VAE objective (Eq.4.4), for which we only optimize the MLP layers, the embedding layer, and the inserted transformer layer, while keeping all other parameters frozen. For the encoder, we use a BERT-small model [Devlin et al., 2019; Turc et al., 2019] and finetune it in the VAE framework. As discussed later in §4.3.4, the tuned encoder can be used to produce the initial $\boldsymbol{z}$ values in the ODE sampler for text editing.

4.3.4 Implementation Details

We discuss more implementation details of the method. Overall, given an arbitrary text corpus (e.g., a set of text from any domain of interest), we first build the VAE by adapting the pretrained LMs as described in §4.3.3. Once the latent space is established, we keep it (including all the VAE components) fixed, and perform compositional text operations in the latent space on the fly.

Acquisition of attribute classifiers We can acquire attribute classifiers $f_i(\boldsymbol{z})$ on the frozen latent space by training using arbitrary datasets with annotations. Specifically, we encode the input text into the latent space with the VAE encoder, and then train the classifier to predict the attribute label given the latent vector. Each classifier, as is built on the semantic latent space, can be trained efficiently with only a small number of examples (e.g., 200 per class). This allows us to acquire a large diversity of classifiers (e.g., sentiment, formality, different keywords) in our experiments (§4.4) using readily-available data from different domains, and flexibly compose them together to perform operations on text in the domain of interest.

Initialization of ODE sampling To sample $\boldsymbol{z}$ with the ODE solver (§4.3.2), we need to specify the initial $\boldsymbol{z}(T)$. For text editing operations (e.g., transferring sentiment from positive to negative) that start with a given text sequence, we initialize $\boldsymbol{z}(T)$ to the latent vector of the given text by the VAE encoder. We show in our experiments that the resulting $\boldsymbol{z}(0)$ samples as the solution of the ODEs can preserve the relevant information in the original text while obtaining the desired target attributes.

For generating new text of target attributes, the normal way is to sample $\boldsymbol{z}(T)$ from the prior Gaussian distribution $\mathcal{N}(\mathbf{0}, I)$. However, due to the inevitable gap between the prior distribution and the learned VAE posterior on $\mathcal{Z}$, such a Gaussian noise sample does not always lead to coherent text outputs. We thus follow [Li et al., 2020; Hu and Li, 2021] to learn a small (single-layer) GAN [Goodfellow et al., 2014] $p_{\text{GAN}}(\boldsymbol{z})$ that simulates the VAE posterior distribution, using all encoded $\boldsymbol{z}$ of real text as the training data. We then generate the initial $\boldsymbol{z}(T)$ from the p_{GAN}.

Sample selection The compact latent space learned by VAE allows us to conveniently create multiple semantically-close variants of a sampled $\boldsymbol{z}(0)$ and pick the best one in terms of certain task criteria. Specifically, we add random Gaussian noise perturbation (with a small variance) to $\boldsymbol{z}(0)$ to get a set of vectors close to $\boldsymbol{z}(0)$ in the latent space and select one from the set. We found the sample perturbation and selection is most useful for operations related to the text content. For example, in text editing (§4.4.2), we pick a vector based on the content preservation (e.g., BLEU with the original text) and attribute accuracy. More details are provided in §4.5.2.

4.4 Experiments

We conduct extensive experiments of composable text controls to show the flexibility and efficiency of LatentOps, including generating new text of compositional attributes (§4.4.1) and editing existing text in terms of desired attributes sequentially or simultaneously (§4.4.2). All code will be released upon acceptance.

Setup We evaluate in two domains, including the Yelp review [Shen et al., 2017] preprocessed by Li et al. [2018b] and the Amazon comment corpus [He and McAuley, 2016]. For each domain, we quickly adapt the GPT2-large to equip with a latent space as described in §4.3.3. The resulting VAE models then serve as the base model, on which we plug in various attribute classifiers for generation and editing. We consider the attributes of *sentiment* (positive, negative), *formality* (formal, informal), and *tense* (pase, present, future). (We also study other attributes related to diverse *keywords*, which we present in §4.5.3). The sentiment/tense classifiers are quickly acquired by training on a small subset of Yelp and Amazon instances (200 labels per class), where the sentiment labels were readily available in the corpus and the tense labels are automatically parsed (§4.5.3). There is no formality information in the Yelp/Amazon corpora, yet the flexibility of LatentOps allows us to acquire the formality classifier using a separate dataset GYAFC [Rao and Tetreault, 2018]. §4.5.3 gives more details of the setup.

4.4.1 Generation with Compositional Attributes

We apply LatentOps to generate new text of arbitrary desired attributes on Yelp domain.

Baselines We compare with the previous plug-and-play text control approaches **PPLM** [Dathathri et al., 2020] and **FUDGE** [Yang and Klein, 2021]. As mentioned earlier, both approaches apply attribute classifiers on the complex sequence space, with an autoregressive LM as a base model. We obtain the base model by finetuning GPT2-large on the above domain corpus (e.g., Yelp). We further compare with an expensive supervised method **GPT2-FT** which finetunes a GPT2-large for *each* combination of attributes. To get the supervised data (§4.5.3), we automatically annotate the domain corpus for formality and tense labels with a trained classifier and tagger, respectively.

Metrics Attribute accuracy is given by a BERT classifier to evaluate the success rate. Perplexity (PPL) is calculated by a GPT2 finetuned on the corresponding domain to measure fluency. We calculate self-BLEU (sBL) to evaluate the diversity. For each case, we sample 150 sequencs to evaluate.

Experimental Results

We list the average results of each combination in Table 4.1. LatentOps achieves observably higher accuracy and diversity, even compared with the fully-supervised method (i.e., GPT2-FT). For fluency, the perplexity of our LatentOps is within a regular interval (the perplexity of human-annotated data is 24.5). However, the baselines obtain excessive perplexity at the expense of diversity.

Table 4.2 shows some generated samples. Ours yields fluent sentences that mostly satisfy the controls. Moreover, GPT2-FT performs similar, although it misses the sub-

Attributes	Methods	Accuracy↑				Fluency↓	Diversity↓
		S	T	F	G-M	PPL	sBL
S	GPT2-FT	0.98	-	-	0.98	10.6	23.8
	PPLM	0.86	-	-	0.86	11.8	31.0
	FUDGE	0.77	-	-	0.77	**10.3**	27.2
	Ours	**0.99**	-	-	**0.99**	30.4	**13.0**
S+T	GPT2-FT	0.98	0.95	-	0.969	9.0	36.8
	PPLM	0.81	0.59	-	0.677	15.7	28.7
	FUDGE	0.67	0.63	-	0.565	**11.0**	35.9
	Ours	**0.98**	**0.93**	-	**0.951**	25.2	**19.7**
S+T+F	GPT2-FT	0.97	0.92	0.87	0.919	10.3	36.8
	PPLM	0.82	0.57	0.56	0.598	17.5	30.5
	FUDGE	0.67	0.64	0.62	0.556	**11.5**	35.9
	Ours	**0.97**	**0.92**	**0.93**	**0.937**	25.8	**21.1**

Table 4.1: Results of generation with compositional attributes. S, T and F stand for sentiment, tense and formality, respectively. G-M is the geometric mean of all accuracy. For reference, the PPL of test data and human-annotated data is 15.9 and 24.5. Since GPT2-FT is a fully-supervised model for reference, we mark the best result **bold** except GPT2-FT.

ject in the second and the third examples. PPLM may fail due to the lack of global concern, e.g., the double negation leads to positive sentiment in the second example. Both PPLM and FUDGE could hardly succeed in all the controls simultaneously since it operates on the sequence space of an autoregressive LM, which is arduous to coordinate the controls. Refer to §4.5.3 for more generated examples and analysis.

Runtime Efficiency

To quantify the computational cost of each method, we evaluate the consumed time for generating 150 examples. For each method, we tested it five times and aver-

Negative + Future + Formal
GPT2-FT:
i will not be back.
would not recommend this location to anyone. [No Subject]
would not recommend them for any jewelry or service. [No Subject]
if i could give this place zero stars, i would.
PPLM:
i could not recommend them at all.
i could not believe this was not good!
this was a big deal, because the food was great.
i could not recommend them.
FUDGE:
not a great pizza to get a great pie! [No Tense]
however, this place is pretty good.
i have never seen anything like these.
will definitely return. [No Subject]
Ours:
i would not believe them to stay .
i will never be back .
i would not recommend her to anyone in the network .
they will not think to contact me for any reason .

Table 4.2: Examples of generation with compositional attributes. We mark failed spans in red.

aged the results as the final result, shown in Table 4.3. Since we sample in the low-dimensional compact latent space, our method is 6.6× faster than FUDGE and 578× faster than PPLM.

Methods	PPLM	FUDGE	Ours
Time (s)	3182 (578×)	36.1 (6.6×)	5.5 (1×)

Table 4.3: Results of generation time of each method.

4.4.2 Text Editing

We evaluate our model's text editing ability on both Yelp and Amazon domains, i.e, changing sentences' sentiment, tense and formality attributes sequentially or altogether.

Baselines Since few previous works can handle the sequential and compositional attributes editing task, we mainly compare with FUDGE [Yang and Klein, 2021]. Moreover, we train three Style Transformer [Dai et al., 2019b] models (for sentiment, tense, and formality, respectively) to sequentially edit the source sentences as a baseline of sequential editing. To show the superiority of our LatentOps, we also conduct text editing with single attribute and compare with several recent state-of-the-art methods (§4.5.3). We adopt the same setting (few-shot) as in §4.4.1 for FUDGE and our LatentOps. It is noteworthy that LatentOps is precisely the same model as in §4.4.1, so it does not require further training.

Metrics Besides success rate and fluency mentioned in §4.4.1, we evaluate the ability of content preservation. Since it is a critical measure lying in the field of text editing, we utilize two metrics: input-BLEU (iBL, BLEU between input and output) and CTC score [Deng et al., 2021] (bi-directional information alignment between input and output). For single attribute setting, we also evaluate reference-BLEU (rBL, BLEU between human-annotated ground truth and output) and perform human evaluations (§4.5.3).

Sequential Editing

In this section, we give the results of sequential editing, whose goal is to edit the given text by changing an attribute each time and keep the main content consistent. We consider the situation that source sentences are with formal manner, positive sentiment and present tense (selected by external classifiers in Yelp), and the goal is to transfer the source sentences to informal manner, negative sentiment and past tense, separately and sequentially. Potential entanglements exist among these attributes, and it is hard to control each attribute independently.

The automatic evaluation results are listed in Table 4.4. LatentOps performs the best on acquiring desired controls and maintaining others and achieves a balanced trade-off among accuracy, content alignment, and fluency. FUDGE fails to introduce the informal manner, while it achieves better formality controls after introducing negative sentiment, showing its deficiency of ability of disentanglement. Furthermore, although FUDGE preserves the most content, it mistakes the core and puts the cart (content) before the horse (accuracy). STrans performs plain overall and cannot guarantee fluency well.

Attributes	Methods	Accuracy			Content↑		Fluency↓
		F	S	T	iBL	CTC	PPL
Informal	FUDGE	0.04	0.06	0.0	**99.4**	0.479	**19.3**
	STrans	0.45	0.14	0.06	65.4	0.470	36.0
	Ours	**0.85**	0.07	0.07	64.2	**0.482**	20.2
+ Negative	FUDGE	0.49	0.35	0.10	**48.6**	0.451	35.0
	STrans	0.38	0.82	0.10	42.4	0.457	39.9
	Ours	**0.75**	**0.92**	0.07	42.1	**0.468**	**28.7**
+ Present	FUDGE	0.48	0.35	0.10	**49.3**	0.452	**30.7**
	STrans	0.36	0.81	0.50	25.6	0.453	45.4
	Ours	**0.61**	**0.83**	**0.74**	20.7	**0.461**	31.5

Table 4.4: Automatic evaluations of sequential editing on Yelp review dataset. F, S and T stand for the accuracy of formality (to informal), sentiment (to negative) and tense (to present), respectively.

We provide some examples in Table 4.5. The formality control of FUDGE makes no effect. Besides, FUDGE would introduce some irrelevant information, e.g., *garlic pizza* and *thing's*. A similar situation exists in STrans, e.g., *ate* and *korean food*. More examples and analysis are in §4.5.3.

Source	the flowers and prices were great .
FUDGE:	
+ informal	the flowers and prices were great. [Formal]
+ negative	garlic pizza and prices were great.
+ present	garlic pizza and prices were great.
STans:	
+ informal	the flowers and prices were great ?
+ negative	the ate and prices were terrible ?
+ present	the ate and prices are terrible ?
Ours:	
+ informal	and the flowers and prices were great !
+ negative	and the flowers and prices were terrible !
+ present	and the flowers and prices are terrible !
Source	best korean food on this side of town .
FUDGE:	
+ informal	best korean food on this side of town. [Formal]
+ negative	thing's best korean food on this side of town.
+ present	thing's best korean food on this side of town. [No Tense]
STans:	
+ informal	best korean food on this side of town korean food . [Formal]
+ negative	only korean food on this side of town korean food .
+ present	only korean food on this side of town korean food . [No Tense]
Ours:	
+ informal	best korean food on this side of town !
+ negative	worst korean food on this side of town !
+ present	this is worst korean food on this side of town !

Table 4.5: Some examples of sequential editing. We mark failed spans in red.

Text Editing with Compositional Attributes

We give the results of text editing with compositional attributes on Yelp, aiming to edit attributes of sentiment and tense of the source sentences. The automatic evaluation results are listed in Table 4.6. LatentOps achieves a higher success rate and content alignment (CTC). FUDGE performs better on iBL and worse on CTC. As demonstrated by Deng et al. [2021], the two-way approach (CTC) is more effective and exhibits a

Methods	Accuracy↑		Content↑		Fluency↓
	Sentiment	Tense	iBL	CTC	PPL
FUDGE	0.36	0.56	**56.5**	0.450	**17.3**
Ours	**0.95**	**0.95**	37.1	**0.465**	30.1

Table 4.6: Automatic evaluation results of text editing with compositional attributes on Yelp review dataset.

higher correlation than single-directional alignment (e.g., BLEU), which is consistent with our observation: FUDGE prefers to generate long sentences that contain the spans in source (raise iBL), but it will also introduce irrelevant information (lower CTC). We give some examples in §4.5.3 to support the claim.

4.4.3 Ablation Study

To clarify the advantage of sampling from ODE, we compare different sampling methods, including Stochastic Gradient Langevin Dynamics (SGLD) and Predictor-Corrector sampler with SDE in §4.5.3.

4.5 In-depth Derivation and Comprehensive Results

4.5.1 Derivation of ODE Formulation

General Form

Let's consider the general diffusion process defined by SDEs in the following form (see more details in Appendix A and D.1 of Song et al. [2021]):

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + \boldsymbol{G}(\boldsymbol{x}, t)d\boldsymbol{w}, \tag{4.9}$$

where $\boldsymbol{f}(\cdot,t):\mathbb{R}^d\rightarrow\mathbb{R}^d$ and $\boldsymbol{G}(\cdot,t):\mathbb{R}^d\rightarrow\mathbb{R}^{d\times d}$. The corresponding reverse-time SDE is derived by Anderson [1982]:

$$\mathrm{d}\boldsymbol{x}=\left\{\boldsymbol{f}(\boldsymbol{x},t)-\nabla_{\boldsymbol{x}}\cdot[\boldsymbol{G}(\boldsymbol{x},t)\boldsymbol{G}(\boldsymbol{x},t)^{\mathrm{T}}]-\boldsymbol{G}(\boldsymbol{x},t)\boldsymbol{G}(\boldsymbol{x},t)^{\mathrm{T}}\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\right\}\mathrm{d}t+\boldsymbol{G}(\boldsymbol{x},t)\mathrm{d}\bar{\boldsymbol{w}}, \tag{4.10}$$

where we refer $\nabla_{\boldsymbol{x}}\cdot\boldsymbol{F}(\boldsymbol{x}):=[\nabla_{\boldsymbol{x}}\cdot\boldsymbol{f}^1(\boldsymbol{x}),...,\nabla_{\boldsymbol{x}}\cdot\boldsymbol{f}^d(\boldsymbol{x})]^{\mathrm{T}}$ for a matrix-valued function $\boldsymbol{F}(\boldsymbol{x}):=[\boldsymbol{f}^1(\boldsymbol{x}),...,\boldsymbol{f}^d(\boldsymbol{x})]^{\mathrm{T}}$, and $\nabla_{\boldsymbol{x}}\cdot\boldsymbol{f}^i(\boldsymbol{x})$ is the Jacobian matrix of $f^i(\boldsymbol{x})$. Then the ODE corresponding to Eq. 4.9 has the following form:

$$\mathrm{d}\boldsymbol{x}=\left\{\boldsymbol{f}(\boldsymbol{x},t)-\frac{1}{2}\nabla_{\boldsymbol{x}}\cdot[\boldsymbol{G}(\boldsymbol{x},t)\boldsymbol{G}(\boldsymbol{x},t)^{\mathrm{T}}]-\frac{1}{2}\boldsymbol{G}(\boldsymbol{x},t)\boldsymbol{G}(\boldsymbol{x},t)^{\mathrm{T}}\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\right\}\mathrm{d}t. \tag{4.11}$$

Derivation of Our ODE

In this work, we adopt the Variance Preserving (VP) SDE [Song et al., 2021] to define the forward diffusion process:

$$\mathrm{d}\boldsymbol{x}=-\frac{1}{2}\beta(t)\boldsymbol{x}\mathrm{d}t+\sqrt{\beta(t)}\mathrm{d}\boldsymbol{w}, \tag{4.12}$$

where the coefficient functions of Eq. 4.9 are $\boldsymbol{f}(\boldsymbol{x},t)=-\frac{1}{2}\beta(t)\boldsymbol{x}\in\mathbb{R}^d$ and $\boldsymbol{G}(\boldsymbol{x},t)=\boldsymbol{G}(t)=\sqrt{\beta(t)}\boldsymbol{I}_d\in\mathbb{R}^{d\times d}$, independent of $\boldsymbol{x}$. Following Eq. 4.10, the corresponding reverse-time SDE is derived as:

$$\begin{aligned}\mathrm{d}\boldsymbol{x}&=\left[-\frac{1}{2}\beta(t)\boldsymbol{x}-\beta(t)\nabla_{\boldsymbol{x}}\cdot\boldsymbol{I}_d-\beta(t)\boldsymbol{I}_d\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\right]\mathrm{d}t+\sqrt{\beta(t)}\boldsymbol{I}_d\mathrm{d}\bar{\boldsymbol{w}}\\&=\left[-\frac{1}{2}\beta(t)\boldsymbol{x}-\beta(t)\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\right]\mathrm{d}t+\sqrt{\beta(t)}\mathrm{d}\bar{\boldsymbol{w}}\\&=-\frac{1}{2}\beta(t)\left[\boldsymbol{x}+2\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\right]\mathrm{d}t+\sqrt{\beta(t)}\mathrm{d}\bar{\boldsymbol{w}},\end{aligned} \tag{4.13}$$

which infers to the Eq. 4.2. Then, we derive the deterministic process (ODE) on the basis of Eq. 4.11:

$$\begin{aligned}
\mathrm{d}\boldsymbol{x} &= \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \frac{1}{2}\beta(t)\nabla_{\boldsymbol{x}} \cdot \boldsymbol{I}_d - \frac{1}{2}\beta(t)\boldsymbol{I}_d\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t \\
&= \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \frac{1}{2}\beta(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t \\
&= -\frac{1}{2}\beta(t)\left[\boldsymbol{x} + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t,
\end{aligned} \tag{4.14}$$

which gives the derivation of Eq. 4.3.

4.5.2 Evaluation of Sample Selection Strategy

As we stated in §4.3.4, we adopt a sample selection strategy for content-related generation tasks (text editing and generation with keywords). Previous works also have similar strategies to improve the generation quality (i.e., PPLM [Dathathri et al., 2020] and FUDGE [Yang and Klein, 2021]).

Since our latent model is trained by VAE objective, a sample $\boldsymbol{x} \in \mathcal{X}$ corresponds to a distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ in $\mathcal{Z}$. Thus, we can search for better output by expanding the search space through sampling $\boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, where $n = 1, ..., N$, and pick the best. Specifically, from ODE sampling, $\boldsymbol{z}(0)$ acts as the mean, and the variance $\boldsymbol{\sigma}^2$ is predefined. We generate $\boldsymbol{z}_n$ by sampling $\boldsymbol{\epsilon}_n$ from standard Gaussian:

$$\boldsymbol{z}_n = \boldsymbol{z}(0) + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_n, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{4.15}$$

We decode each $\boldsymbol{z}_n$ and pick the best one according to the criterion of the task. We prefer the output that conforms to the desired attribute and achieves a high BLEU score with the source text for the text editing task. We want the output that contains the desired keyword or its variants for the generation with keywords.

In our experiments (text editing and generation with keywords), we set $N = 20$ as the default. To better demonstrate the strategy's improvement, we provide the quantitative and qualitative results towards different N.

We follow the same setting of text editing with single attribute on Yelp (§4.5.3). The automatic evaluation results are shown in Table 4.7. As N increases, all the metrics get improved. To reflect the trend of change in accuracy and content preservation, we plot Figure 4.3, which indicates that large N gives better accuracy and better input-BLEU.

We also provide some examples in Table 4.8 and Table 4.9. One observation is that all the outputs from the same source sequence describe similar scenarios but

N	Accuracy↑	Content↑			Fluency↓
	Sentiment	iBL	rBL	CTC	PPL
2	0.75	51.1	21.4	0.4737	26.3
4	0.82	50.6	22.0	0.4729	26.7
6	0.89	49.6	22.3	0.4729	26.2
8	0.9	50.5	22.2	0.4732	25.9
10	0.92	50.8	23.1	0.4730	26.2
12	0.93	51.4	23.2	0.4733	26.1
14	<u>0.94</u>	51.4	23.0	0.4732	26.9
16	<u>0.94</u>	52.4	23.4	0.4737	<u>25.9</u>
18	**0.95**	<u>52.6</u>	<u>23.6</u>	<u>0.4739</u>	**25.8**
20	**0.95**	**54.0**	24.2	**0.4743**	<u>25.9</u>

Table 4.7: Automatic evaluation results towards to different N on Yelp review dataset. We mark the best **bold** and the second best <u>underline</u>.

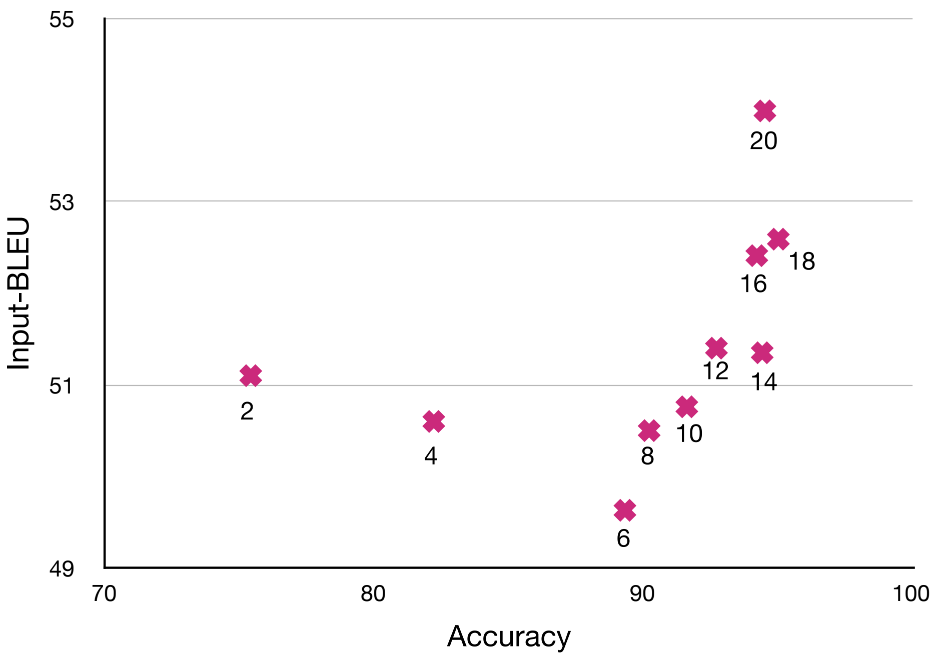

Figure 4.3: The trend of change of accuracy and input-BLEU as N increases. The digit below each data point represents the corresponding N.

slightly differ in expression. Thus, we can select the most suitable expression based on predefined rules.

Source Target	there is definitely not enough room in that part of the venue . there is so much room in that part of the venue
	there is definitely plenty of room in that perfect location . there is definitely no room enough in that venue to be the best part . there is definitely plenty of room right in that venue . there is definitely plenty of room right in the venue that needs . there is definitely plenty of room right in that venue . there is definitely enough room that can be right in the venue . there is definitely nothing better in room for that type of venue . there is definitely plenty of room in the right venue for that level . there is definitely nothing better in that room style of place . there is definitely a good room inside that best of all need in space . there is definitely plenty of room in the right level that is appropriate . there is definitely enough room in that right part of the venue . there is definitely plenty of room right in the deck that is needed . there is definitely enough room in that good atmosphere . there is definitely plenty of room in the right area , which is comfortable . there is definitely plenty of room in that perfect state of the place . there is definitely plenty of room that ideal in the location . there is definitely enough room in that perfect venue to all . there is definitely plenty of room in the right venue as well . there is definitely plenty of room available in the overall venue , too .
Source Target	it is n't terrible , but it is n't very good either . it is n't perfect , but it is very good .
	it is n't terrible , but it is very good also ! it is very good , but it does n't even look great ! it is n't terrible , but it is very good and definitely is good ! it is n't great , but it is definitely very good ! it is n't terrible , it is good and the menu is definitely great ! it is n't terrible , but it is n't very good either . it is n't terrible , but it is very good also . it is n't terrible , but it is very good also ! it is n't terrible , but it is definitely very good ! it is very good , and it is n't terrible either . it is n't terrible , but it is very good and well made ! it is very good , but it 's not really great either . it is n't terrible , but it is very good and well worth it . it is n't terrible , but it is definitely very good and good ! it is n't terrible , but it is very good also ! it is n't terrible , but it is very good and definitely is great ! it is n't terrible , but it is very good also . it is n't terrible , but it is n't very good either . it is n't terrible , but it is very good also . it is n't terrible , but it is very good and always great !

Table 4.8: Examples of sample selection strategy ($N = 20$).

Source	the food was pretty bad , i would not go there again .
Target	the food was great, i would go there again.
	he food was pretty good , i would go there again .
	the food was pretty good , i would def go there again !
	the food was pretty good , i would go again !
	the food was pretty good , i would go there again !
	the food was pretty good , i would definitely go there again .
	the food was pretty good , i would go back there again .
	the food was pretty good , i would definitely go back again .
	the food was pretty good , i would definitely go there again !
	the food was pretty good , i would definitely go there again .
	the food was pretty good , i would always go there again .
	the food was pretty good , i would go there again .
	the food was pretty good , i would not go there again .
	the food was pretty good , i would go there again .
	the food was pretty good , i would go back there again .
	the food was pretty good , i would go there again .
	the food was pretty good , i would definitely go there again !
	the food was pretty good , i would not go there again .
	the food was pretty good , i would definitely go there again .
	the food was pretty good , i would definitely go back again .
	the food was pretty good , i would go here again .

Table 4.9: Examples of sample selection strategy ($N = 20$).

4.5.3 More Details and Results of Experiments

In this section, we provide more details and results of the experiments (§4.4).

Setup

The Yelp dataset and Amazon dataset contain 443K/4K/1K and 555K/2K/1K sentences as train/dev/test sets, respectively. Since Yelp and Amazon datasets[2][3] are mainly developed for sentiment usage, we annotate them with a POS tagger to get the tense attribute to test the ability of our model that can be extended to an arbitrary number of attributes. Besides, we also use GYAFC dataset [Rao and Tetreault, 2018] to include the formality attribute. Note that the GYAFC dataset has somewhat different domains from Yelp/Amazon, which can be used to test our model's out-of-domain generalization ability. All the datasets are in English.

We adopt BERT-small[4] and GPT2-large[5] as the encoder and decoder of our latent model, respectively. The training paradigm follows §4.3.4, and some training tricks [Li et al., 2020] (i.e., cyclical schedule for KL weight and KL thresholding scheme) are applied to stabilize the training of the latent model. All the attributes are listed in Table 4.10. All the models are trained and tested on a single Tesla V100 DGXS with 32 GB memory. Input-BLEU, reference-BLEU and self-BLEU are implemented by nltk [Bird et al., 2009] package.

For the operator (classifier) $f_i(\boldsymbol{z})$, we adopt a four-layer MLP as the network architecture as shown in Table 4.11. Since the number of trainable parameters of the classifier is small, it is rapid to train and sample.

[2] https://github.com/lijuncen/Sentiment-and-Style-Transfer
[3] The datasets are distributed under CC BY-SA 4.0 license.
[4] The BERT model follows the Apache 2.0 License.
[5] The GPT2 model follows the MIT License.

Style	Attributes	Dataset
Sentiment	Positive / Negative	Yelp, Amazon
Tense	Future / Present / Past	Yelp
Keywords	Existence / No Existence	Yelp
Formality	Formal / Informal	GYAFC

Table 4.10: All attributes and the corresponding dataset are used in our experiments.

Input	Layer 1	Layer 2	Layer 3	Layer 4
$z \in \mathbb{R}^{64}$	Linear 43, LeakyReLU	Linear 22, LeakyReLU	Linear 2, LeakyReLU	Linear #logits

Table 4.11: The architecture of the attribute classifier.

Generation with Compositional Attributes

The section is a supplement of §4.4.1, we give more details of experimental configuration, generated examples and discussion.

More Details of Baselines We compare our method with PPLM [Dathathri et al., 2020], FUDGE [Yang and Klein, 2021], and a finetuned GPT2-large [Radford et al., 2019b]. PPLM and FUDGE are plug-and-play controllable generation approaches on top of an autoregressive LM as the base model. For fair comparison (§4.3.3), we obtain the base model by finetuning the embedding layer and the first transformer layer of pretrained GPT2-large on the Yelp review dataset with unlabeled data. All the classifiers/discriminators of PPLM, FUDGE and our LatentOps are trained by a small subset of the original dataset (200 labeled data instances per class).

PPLM requires a discriminator attribute model (or bag-of-words attribute models) learned from a pretrained LM's top-level hidden layer. At decoding, PPLM modifies the states toward the increasing probability of the desired attribute via gradient ascent. We only consider the discriminator attribute model, which is consistent with other baselines and ours. We follow the default setting of PPLM, and for each attribute, we train a single layer MLP as the discriminator.

FUDGE has a discriminator that takes in a prefix sequence and predicts whether the generated sequence would meet the conditions. FUDGE could control text generation by directly modifying the probabilities of the pretrained LM by the discriminator output. We follow the architecture of FUDGE and train a discriminator for each attribute. Furthermore, we tune the λ parameter of FUDGE which is a weight that controls how much the probabilities of the pretrained LM are adjusted by the discriminator, and we find λ=10 yields the best results. We follow the default setting of FUDGE, and for each attribute, we train a three-layer LSTM followed by a Linear as the discriminator.

GPT2-FT is a finetuned GPT2-large model that is a conditional language model, not plug-and-play. Specifically, we train an external classifier for the out-of-domain attribute (i.e., formality) to annotate all the data in Yelp. For tense, we use POS tagging to annotate the data automatically. Then we finetune the embedding layer and the first layer of GPT2-large by the labeled data. Since GPT2-FT is fully-supervised and not plug-and-play, it is not comparable with other baselines and ours, and we only use it for reference.

More Discussion of Generation with Compositional Attributes

Discussion of Quantitative Results As we state in §4.4.1, our method is superior to baselines. We want to discuss the results in Table 4.1.

For success rate, our method dramatically outperforms FUDGE and PPLM as expected since both control the text by modifying the outputs (hidden states and probabilities) of PLM, which includes the token-level feature and lacks the sentence-level semantic feature. On the contrary, our method controls the attributes by operating the sentence-level latent vector, which is more suitable.

For diversity, since our method bilaterally connects the discrete data space with continuous latent space, which is more flexible to sample, ours gains obvious superiority in diversity. Conversely, PLMs like GPT2, which is the basis of PPLM and FUDGE, are naturally short of the ability to generate diverse texts. They generate diverse texts by adopting other decoding methods (like top-k), which results in the low diversity of the baselines.

For fluency, we calculate the perplexity given by a finetuned GPT2, which processes the same architecture and training data of PPLM and FUDGE, so naturally, they can achieve better perplexity even compared to the perplexity of test data and human-annotated data. Moreover, our method only requires an Extra Adapter to guide the fixed GPT2, and our fluency is in a regular interval, a little higher than the perplexity of human-annotated data.

Since GPT2-FT is trained with full joint labels (all the data has all three attribute labels), it can achieve a reasonable success rate, and ours is comparable. Moreover, consistent with PPLM and FUDGE, GPT2-FT can achieve good perplexity but poor diversity due to the sampling method.

Discussion of Qualitative Results We provide some generated examples in Table 4.12 and Table 4.13 to raise a more direct comparison. Consistent with the quantitative results, it is difficult for FUDGE to control all the desired attributes successfully, although GPT2-FT and ours perform well. For diversity, it is evident that FUDGE and GPT2-FT prefer to generate short sentences containing very little information. Some words appear highly, yet ours gives a more diverse description. Regarding fluency, since FUDGE and GPT2-FT tend to generate simple sentences, they can obtain better perplexity readily. However, ours is inclined to generate more informative sentences. In conclusion, there is a trade-off between diversity and fluency.

It can be handled well by ours, but for the baselines, they pursue fluency too much and lose diversity.

Positive + Present + Formal	**Negative + Past + Inormal**
GPT2-FT: the staff is friendly and helpful. i love it here. [Informal] this is the place to go for traditional chinese food. highly recommend them. [Informal] the menu is small but very nice. it's a great place. i highly recommend this place.	GPT2-FT: didn't bother with the food and just walked out. just not a good place for me. [No Tense] not a fan of this place. [No Tense] just not good. [No Tense] horrible! [No Tense] oh and the cake was way too salty. but we didn't even finish it.
PPLM: i love this store and the service is always friendly and courteous. the staff was so friendly & helpful![Informal] the place is clean. the best french bakery i have ever been to in las vegas! this place was a gem! she does love to make suggestions and i appreciate that. they also always remember us and always always get us right in and always have good prices.	PPLM: i ordered delivery... what? great service. [No Tense] this place was terrible! the service was horrible horrible horrible! i ordered the ribs and brisket tacos and it was very bland. [Formal] the staff was very apologetic and apologetic and refund my $ _num_ for the oil change [Formal] i ordered pizza and wings from brooklyn's and they were all out of ranch. [Formal]
FUDGE: great for breakfast or a nice lunch. [Informal] great location. [Informal] their staff is friendly, professional, and the facility is clean and comfortable. great. [Informal & No Tense] great place for lunch or a date. [No Tense] great place! [Informal & No Tense] great food. [Informal & No Tense]	FUDGE: came to phoenix from new jersey last weekend...! food was ok, but service was terrible! usually the service was good and the food was good no complaints. food was ok but our waiter was awful. c was amazing. c was so good and i highly recommend. ch was the only reason i stayed for the night.
Ours: the food is clearly great , as they are always tasty . they are really knowledgeable , what draws me . the shop is authentic , their hair is great . the food is always unique with well spiced . that is a great form of customer service . they have very professional people who are worth their service . i love living there as does my clients .	Ours: everything was a bit cold but anyways , i ordered them ! anyway i had the worse experience ! looked like i was n't even paid this money ! (had no job in _*num*_ months from cali .) i waited at the room & got _*num*_ people yelling ? (i didnt get this at all times) they had me cold a lot !

Table 4.12: More examples of generation with compositional attributes. We mark failed spans in red.

Negative + Future + Formal	**Positive + Past + Informal**
GPT2-FT: i will not be back. would not recommend this location to anyone. would not recommend them for any jewelry or service. if i could give this place zero stars, i would. if i could give no stars, i would. i would not recommend this place to anyone. i can not get my medication on time.	GPT2-FT: good prices too! [No Tense] i even liked the cheese curds.... hands down the best sushi i've had in a while. just a great shop! [No Tense] my friend had a good time. got ta love that! really good service, super fast and friendly. [No Tense]
PPLM: i could not recommend them at all. i could not believe this was not good! this was a big deal, because the food was great. i could not recommend them. i will not be back. the food was mediocre. they were not.	PPLM: i ordered a great deal at a very good sushi restaurant tonight. [Formal] it is light and airy and has very few after tastes of smoke or heat. i loved it so much i had to get the other salad! the staff at my table had the best service ever! we've had some really great ones too. i love everything and would highly recommend! they did a fabulous job of putting me on a diet for the first time in my life! [Formal]
FUDGE: not a great pizza to get a great pie! [No Tense] however, this place is pretty good. i have never seen anything like these. will definitely return. i would have loved to have a nice lunch here. they don't have any of the ingredients they should. do not go here for the food.	FUDGE: thanks was definitely great! went and spent the whole night here and had a blast! she loved the food and service! went and the food was good, nothing special. he was friendly, knowledgeable and very helpful! great beer was amazing! went on to eat and was very disappointed with our food!
Ours: i would not believe them to stay . i will never be back . i would not recommend her to anyone in the network . they will not think to contact me for any reason . i should not risk coming to this establishment . i would not waste more time in henderson . i doubt i would 've ever been to this airline .	Ours: everything was hot and incredibly good ! plus they had a great and fresh meal here ! fresh mozzarella was great in general ! the veggies and omelette were great ! great service and enjoyed our out day meal i ended up getting a great meal (i loved it !) (she got a job for me !

Table 4.13: More examples of generation with compositional attributes. We mark failed spans in red.

Results of Generation with Compositional Attributes and Keywords We regard keywords as an attribute of the text sequence. To prepare the data, we extract all verbs, nouns, and variants that appeared in the Yelp review dataset, filter out the sentiment-related words[6], and construct the training data. Then, we obtain 613 keywords listed in Table 4.15 and Table 4.16. We treat each keyword (e.g., *have*) and their variants (e.g., *had* or *has*) equally without discrimination. Moreover, for each keyword, we randomly select 220 sentences where the keyword exists and 220 sentences that do not include the keyword as the training data (200) and test data (20). Since we have 3,678 combinations of keyword, sentiment and tense, we adopt a pretrained GPT2 base model as the decoder to accelerate the process.

We conduct the experiments of single keyword and keyword combining with other attributes (sentiment and tense). We first give the automatic evaluation results in Table 4.14. We list the average results of each combination of keywords, sentiment and tense. All success rates, diversity and fluency, are at a high level. To make the results more intuitive, we also give some generated examples in Table 4.17 and Table 4.18.

Attributes	Accuracy↑				Fluency↓	Diversity↓
	Keyword	Sentiment	Tense	G-Mean	PPL	sBL
Keyword	0.98	-	-	0.98	21.7	10.6
+ Sentiment	0.94	0.96	-	0.95	21.3	10.8
+ Tense	0.93	0.9	0.93	0.92	19.7	10.9

Table 4.14: Results of generation with compositional attributes and keywords.

Results of Generation with Single Attribute Table 4.19 gives the results of single-attribute conditional generation. Our method dramatically outperforms PPLM and

[6] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

FUDGE for all attributes on the accuracy, exceeding 94%. The diversity and fluency of our method are consistent with multi-attribute results.

Text Editing

The section is a supplement of §4.4.2, we give more details of experimental configuration, generated examples and discussion.

More Details of Baselines For text editing, we experiment with three settings–sequential attribute editing, compositional attributes editing and single attribute editing.

We compare with several recent state-of-the-art methods: B-GST [Sudhakar et al., 2019], Style Transformer (STrans) [Dai et al., 2019b], DiRR [Liu et al., 2021], Tag&Gen (T&G) [Madaan et al., 2020], and fine-grained style transfer (FGST) [Liu et al., 2020]. The outputs of baselines are obtained from their official repositories except for FUDGE. Since FUDGE relies on a PLM, we finetune a GPT2 as a reconstruction model as the base model.

FUDGE is the sole model that could handle compositional attributes. Therefore, we compare with FUDGE in the compositional attributes setting. Furthermore, we tune the λ parameter of FUDGE which is a weight that controls how much the probabilities of the pretrained LM are adjusted by the discriminator, and we find λ=100 yields the best results. We compare with all baselines in the single attribute setting.

Examples of Sequential Editing We provide more examples of the Sequential Editing (§4.4.2) experiment in Table 4.20 and Table 4.21, where the two examples in Table 4.20 are the same as in 4.5. Our method can sequentially edit the source text to desired attributes more smoothly and consistently.

In the first example in Table 4.20, FUDGE fails on all three edits, Style Transformer introduces *ate*, which leads to grammatical mistakes and loss of critical information (*flowers*). Our method can edit the source text step-by-step successfully.

In the second example in Table 4.20, FUDGE fails all edits again and introduces irrelevant information (*thing's*). Furthermore, Style Transformer nearly fails in all edits. Our method could generate both fluent and content-relevant sentences.

In the first example in Table 4.21, we consider editing the source to formal, positive and past. FUDGE and Style Transformer only succeed in introducing the positive sentiment, and FUDGE also introduces some redundant information (*to get away from the strip*). Ours first extends the source to be formal, then changes the sentiment (*horrible* to *amazing*) and tense (*is* to *was*), sequentially.

In the last example in Table 4.21, FUDGE fails all edits. Although Style Transformer succeeds in sentiment transfer, the generated sentence is not grammatically correct. Ours could generate eligible and fluent sentences.

Initia	Keywords
a	accommodate add afternoon agree airport ambiance ambience amount animal answer anyone anything apartment apologize apology appetizer appointment area arizona arrive art ask atmosphere attention attitude auto average avoid az
b	baby back bacon bag bagel bakery bar bartender base bathroom bbq bean beat become bed beef beer begin believe bell bike bill birthday biscuit bit bite book bottle bowl box boy boyfriend bread breakfast bring brunch buck buffet building bun burger burrito business butter buy
c	cab cafe cake call car card care carry case cash cashier center chain chair chance change charge charlotte check cheese chef chicken child chili chip chocolate choice choose city class cleaning close club cocktail coffee color combo come company condition consider contact continue cook cooky corn cost counter couple coupon course cover crab crave cream credit crew crispy crowd crust cup curry customer cut
d	date daughter day deal dealership decide decor deli deliver delivery dentist department deserve desk dessert detail diner dining dinner dip discount dish do doctor dog dollar donut door downtown dress dressing drink drive driver drop
e	eat egg employee enchilada end entree environment establishment evening event everyone everything expect expectation experience explain eye
f	face facility fact family fan fee feel feeling felt fill find finish fish fit fix flavor flight floor flower folk follow food foot forget friday friend front fruit fry furniture future
g	game garden get gift girl give glass go god grab greet grill grocery ground group guess guest guy gym gyro
h	hair haircut half hand handle happen have head hear heart help hit hold hole home homemade honey hope hospital hostess hotel hour house husband
i	ice idea include ingredient inside item
j	job joint juicy
k	keep kid kind kitchen know
l	lady leave let lettuce level life light line list listen live lobster location look lot lunch

Table 4.15: All keywords. Sort in alphabetical order.

Initia	Keywords
m	mac machine madison make mall man management manager manicure manner margarita mark market massage matter meal mean meat meatball medium meet melt member mention menu mile min mind mine minute mix mom money month morning mouth move movie mushroom music
n	nail name need neighborhood night none noodle notch nothing notice number nurse
o	occasion offer office oil ok okay omelet one onion online open opinion option orange order organize others overcook overprice own owner
p	pack pad pancake park parking part party pass pasta patio pay pedicure people pepper person pet phoenix phone pick picture pie piece pittsburgh pizza place plan plate play please plenty point pool pork portion potato practice prepare price pricing process produce product provide purchase put
q	quality question quick quote
r	ranch rate rating read reason receive refill relax remember rent repair replace request reservation resort rest restaurant result return review rib rice ride ring rock roll room run rush
s	salad sale salmon salon salsa salt salty sandwich saturday sauce sausage save saw say schedule school scottsdale seafood season seat seating section see seem selection sell send sense serve server service set share shoe shop shopping shot show shrimp side sign sit size slice soda someone something son sound soup space speak special spend spice spicy spinach sport spot spring staff stand standard star starbucks start state station stay steak step stick stock stop store story street strip stuff style stylist sub suggest summer sunday suppose surprise sushi
t	table taco take talk taste tasty tea team tech tell thai thanks theater thing think throw time tip tire toast today tomato ton tonight topping tortilla touch town treat trip try tuna turn tv type
u	understand update use
v	valley value vega vegetable veggie vehicle venue vet vibe view visit
w	waffle wait waiter waitress walk wall want wash watch water way wedding week weekend while wife window wine wing wish woman word worker world wrap write
y	year yelp yesterday yummy

Table 4.16: All keywords. Sort in alphabetical order.

Keyword: *expectation* the prices were excellent and exceeded our expectations . five stars , affordable and reasonable pricing exceeded my expectations . i 've had four peaks meal from my expectations and i have not disappointed . you are crazy close to my expectations ! the flavors have never been above & beyond expectations .
Keyword: *expectation* + **Sentiment**: Negative the appetizers were completely lower expectations . i would give this restaurant _*num*_ zero expectations in terms of our entrees . it was n't that impressive and _*num*_ declined my expectations . there were zero expectations . but my expectations were lower than zero stars .
Keyword: *expectation* + **Sentiment**: Negative + **Tense**: Past there were so low expectations throughout the end . the food was ok , but my expectations were high to top notch . during the event we were already disappointed with the expectations . we arrived _*num*_ months ago and my expectation was overcharged . again , the initial estimate of course had not gotten my expectations and declined .
Keyword: *expectation* + **Sentiment**: Negative + **Tense**: Present the prices are really low and restaurants are not above expectations . there is almost no flavor in my expectations . the chips and salsa are far below their expectations and lack of manners . it 's about the expectations lower than zero . the food in american restaurants do not exceed your expectations .
Keyword: *expectation* + **Sentiment**: Negative + **Tense**: Future i would not come back to any expectations of this restaurant . it would n't be exceeded my expectations at any point . i would n't want you to have any expectations in this hotel . honestly i would n't have lower expectations before . i would not expect superior from my expectation .

Table 4.17: Examples of generation with compositional attributes with keywords (*expectation* and *accommodate*).We mark the spans that conform to desired attributes in blue.

Keyword: *accommodate*
staff was nice and accommodating a timely manner .
he is always nice and accommodating .
the service is wonderful and the facility is clean and accommodating .
nicely crowded , along with a great accommodating staff !
she is friendly and willing to accommodate any type of questions .
Keyword: *accommodate* + **Sentiment**: Positive
staff is very nice and the servers are friendly and accommodating .
everyone was very friendly and accommodating with a ton of energy !
tamara was extremely nice and accommodating .
everyone seemed to talk with accommodating .
he made a wonderful massage to accommodate my kids .
Keyword: *accommodate* + **Sentiment**: Positive + **Tense**: Past
they were really nice and made to accommodate me with a great energy .
the everyone was very nice and the hospitality was accommodating as well !
the whole family was accommodating and we enjoyed the round !
the staff was always friendly and accommodating with great suggestions .
thanks , the hostess was extremely helpful and accommodating .
Keyword: *accommodate* + **Sentiment**: Positive + **Tense**: Present
they are friendly and helpful , and the pricing is easy to accommodate .
the staff is amazing and very accommodating and the owners are wonderful .
everyone is super nice and accommodating !
the servers are always accommodating and helpful !
the venue is quite accommodating , and a great happy atmosphere .
Keyword: *accommodate* + **Sentiment**: Positive + **Tense**: Future
they will definitely stay close to accommodate us !
they would very reasonable to accommodate you in any condition !
hopefully , they will definitely be accommodated with our family !
they would be able to accommodate you at any location .
i would definitely recommend this firm to accommodate us !

Table 4.18: Examples of generation with compositional attributes with keywords (*expectation* and *accommodate*).We mark the spans that conform to desired attributes in blue.

Attributes	Methods	Accuracy↑	LogVar↓	Fluency (PPL)↓	Diversity (sBL) ↓
Sentiment	GPT2-FT	0.98	-11.31	10.6	23.8
	PPLM	0.86	-4.68	11.8	31.0
	FUDGE	0.77	-2.97	**10.3**	27.2
	Ours	**0.99**	**-Inf**	30.4	**13.0**
Tense	GPT2-FT	0.97	-9.33	10.0	31.0
	PPLM	0.6	-3.30	13.9	27.8
	FUDGE	0.77	-3.11	**10.9**	37.6
	Ours	**0.96**	**-6.8**	36.7	**9.5**
Formality	GPT2-FT	0.88	-5.75	14.9	18.0
	PPLM	0.62	-2.43	14.8	24.8
	FUDGE	0.59	-2.16	**11.2**	28.6
	Ours	**0.97**	**-7.82**	36.3	**12.0**

Table 4.19: Automatic evaluation results of generation with single attribute. We show the natural logarithm of variance (LogVar) of accuracy, since the original scale is too small for demonstration.

Source	the flowers and prices were great .
FUDGE:	
+ informal	the flowers and prices were great. [Formal]
+ negative	garlic pizza and prices were great.
+ present	garlic pizza and prices were great.
STans:	
+ informal	the flowers and prices were great ?
+ negative	the ate and prices were terrible ?
+ present	the ate and prices are terrible ?
Ours:	
+ informal	and the flowers and prices were great !
+ negative	and the flowers and prices were terrible !
+ present	and the flowers and prices are terrible !
Source	best korean food on this side of town .
FUDGE:	
+ informal	best korean food on this side of town. [Formal]
+ negative	thing’s best korean food on this side of town.
+ present	thing’s best korean food on this side of town. [No Tense]
STans:	
+ informal	best korean food on this side of town korean food . [Formal]
+ negative	only korean food on this side of town korean food .
+ present	only korean food on this side of town korean food . [No Tense]
Ours:	
+ informal	best korean food on this side of town !
+ negative	worst korean food on this side of town !
+ present	this is worst korean food on this side of town !

Table 4.20: Examples of sequential editing. We mark failed spans in red.

Source	horrible .
FUDGE:	
+ formal	horrible! [Informal]
+ positive	great place to get away from the strip.
+ past	great place to get away from the strip. [No Tense]
STrans:	
+ formal	horrible . [Informal]
+ positive	wonderful .
+ past	wonderful .[No Tense]
Ours:	
+ formal	service is completely horrible .
+ positive	service is completely amazing .
+ past	service was completely amazing .
Source	**it is a garbage , and nobody does really care !**
FUDGE:	
+ informal	it is a garbage , and nobody does really care ! [Formal]
+ positive	it is always a garbage , and nobody does really care !
+ future	it is always a garbage , and nobody does really care !
STrans:	
+ informal	it is a garbage , and nobody does really care ! [Formal]
+ positive	it is a smile , and high does really care !
+ future	it is a smile , and high does really care !
Ours:	
+ informal	(it is garbage services ... no crap !
+ positive	(the delivery service is excellent !)
+ future	it is the first delivery service i will get !

Table 4.21: Examples of sequential editing. We mark failed spans in red.

Examples of Text Editing with Compositional Attributes We provide some examples of Text Editing with Compositional Attributes (§4.4.2) in Table 4.22 and Table 4.23.

Source	so basically tasted watered down .
Human	it didn't taste watered down at all.
FUDGE	once every couple months, we get a new car - so basically tasted watered down.
+ Past	such basically tasted watered down.
+ Present	such basically tasted watered down.
+ Future	very watered down.
Ours	so basically tasted delicious .
+ Past	so nicely tasted watered down .
+ Present	so basically tastes delicious .
+ Future	so basically you will be satisfied .
Source	it is n't terrible , but it is n't very good either .
Human	it is n't perfect , but it is very good .
FUDGE	its good, but it isn't very good either.
+ Past	whether on vacation or in the car, this hotel isn't terrible, but it isn't
+ Present	whether good the food isn't terrible, but it isn't very good either. good good
+ Future	several locations aren't terrible, but it is good very good good great!
Ours	it is n't terrible , but it is very good also .
+ Past	it was n't terrible , but it was very good and quick !
+ Present	it is n't terrible , but it is very good also .
+ Future	it is n't terrible , but it would definitely be very good !

Table 4.22: Examples of text editing with compositional attributes (sentiment and tense) on the Yelp review dataset. Human is the human-annotated reference for sentiment transfer. We mark the failed spans red and successful spans blue.

Source	anyway , we got our coffee and will not return to this location .
Human	we got coffee and we'll think about going back
FUDGE	exactly zero stars for any way, we got our coffee and will not return to this location.
+ Past	once our coffee and will not return to this location.
+ Present	once, we got our coffee and will not return to this location.
+ Future	once again, we got our coffee and will not return to this location.
Ours	anyway , we got our coffee and will always return to this location .
+ Past	anyway , we got our coffee and delivered to this friendly location .
+ Present	anyway , we love our coffee and this location has to be found .
+ Future	anyway , we got our coffee and will continue to return to this location .
Source	this place is a terrible place to live !
Human	this place is a great place to live !
FUDGE	great place to live!
+ Past	great food and terrible service! [No Tense]
+ Present	great place to live! [No Tense]
+ Future	great place to live! [No Tense]
Ours	this place is a great place to live !
+ Past	this place was a great place to live !
+ Present	this place is a great place to live !
+ Future	this place would have a great place to live !

Table 4.23: Examples of text editing with compositional attributes (sentiment and tense) on the Yelp review dataset. Human is the human-annotated reference for sentiment transfer. We mark the failed spans red and successful spans blue.

Results of Text Editing with Single Attribute We conduct text editing with a single attribute on both the Yelp review dataset and the Amazon comment corpus. Since both Yelp and Amazon provide 1000 human-annotated sentences, we also calculate reference-BLEU (rBL, BLEU score between output and human-annotated sentences).

The automatic evaluation results are in Table 4.24. Given a pretrained latent model, ours only requires training a classifier of 3.7K parameters and achieves competitive results compared with the strong baselines of many more parameters. Regarding the success rate, our method is in the premier league compared to the methods trained with full labeled data. In respect of content preservation, DiRR distinctly outperforms others, since DiRR processes 1.5B trainable parameters and is trained on the full labeled data ($\sim$440K training data), so big data and big models lead to better performance. However, although we follow the few-shot setting (400 training data), ours also performs well in preserving content. Compared with strong baselines, our method achieves competitive results at fluency and input-output alignment (CTC).

We also perform human evaluations on Yelp to further measure the transfer quality. Three people with related experience are invited to score the generated sentences (1 for low quality and 4 for high quality). We then average the scores as the final human evaluation results. As the human evaluation results are shown in Table 4.24, our LatentOps performs the best. Some generated examples are provided in Table 4.25, Table 4.26, Table 4.27 and Table 4.28 to further demonstrate the superiority of our method. One observation is that our method could focus more on logicality and adopt words appropriate to the context.

Ablation Study: Comparison with SGLD and SDE

In order to show the superiority of the ODE sampler introduced in §4.3.2, we compare with Stochastic Gradient Langevin Dynamics (SGLD) and Predictor-Corrector sam-

Methods	Accuracy↑	Content↑			Fluency↓	#Params	#Data
	Sentiment	iBL	rBL	CTC	PPL		
Source	0.27	100	31.4	0.500	15.9	-	-
Human	0.82	31.9	100	0.463	24.5	-	-
B-GST	0.81	31.8	16.3	0.473	39.5	111M	Full-data
STrans	0.91	53.2	<u>24.5</u>	0.469	41.0	17M	
DiRR	**0.96**	**61.5**	**29.8**	**0.480**	<u>23.9</u>	1.5B	
T&G	0.88	47.6	21.8	0.466	24.3	63M	
FGST	0.90	13.2	7.6	0.450	**9.3**	26M	
FUDGE	0.40	<u>57.0</u>	18.0	0.456	39.3	16.4M	**Few-shot**
Ours	<u>0.95</u>	54.0	24.3	<u>0.474</u>	25.9	**3.7K**	
Source	0.14	100	49.4	0.425	26.4	-	-
Human	0.52	49.7	100	0.422	47.2	-	-
B-GST	0.62	52.3	28.5	<u>0.425</u>	<u>27.7</u>	111M	Full-data
DiRR	0.60	<u>68.7</u>	**38.2**	0.424	32.5	1.5B	
T&G	0.65	68.6	<u>35.4</u>	0.423	40.9	63M	
FGST	**0.83**	21.9	14.0	**0.427**	**13.6**	26M	
FUDGE	0.20	**70.5**	35.1	0.415	49.5	16.4M	**Few-shot**
Ours	<u>0.72</u>	53.3	28.1	0.423	44.1	**3.7K**	

B-GST	STrans	DiRR	T&G	FGST	FUDGE	Ours
2.03	2.20	3.13	2.20	1.60	1.20	**3.27**

Table 4.24: Automatic evaluations of text editing with single attribute on Yelp (top) and Amazon (middle) dataset. We mark the number of trainable parameters as #Params and the scale of labeled data in training as #Data. Human evaluation (bottom) statistics on Yelp.

Source	so basically tasted watered down .
Human	it didn’t taste watered down at all.
B-GST	so basically tasted delicious .
STrans	so basically really clean and comfortable .
DiRR	so basically tastes delicious .
T&G	everything tasted fresh and tasted delicious .
FGST	everything tasted fresh and tasted like watered down .
FUDGE	once every couple months, we get a new car - so basically tasted watered down.
Ours	so basically tasted delicious .
Source	it is n’t terrible , but it is n’t very good either .
Human	it is n’t perfect , but it is very good .
B-GST	best indian food in whole of pittsburgh .
STrans	it is n’t great , but it is very good atmosphere .
DiRR	it is great , but it is very good either .
T&G	it is n’t great , but it is n’t very good .
FGST	the food is n’t very good , but it is n’t great either .
FUDGE	its good, but it isn’t very good either.
Ours	it is n’t terrible , but it is very good also .
Source	anyway , we got our coffee and will not return to this location .
Human	we got coffee and we’ll think about going back
B-GST	”got our tickets
STrans	anyway , we got our coffee and will definitely return to this location .
DiRR	anyway , we got our coffee and will definitely return to this location .
T&G	anyway , we got our coffee and we will definitely return in town .
FGST	we will return to this location again , and the coffee was great .
FUDGE	exactly zero stars for any way, we got our coffee and will not return to this location.
Ours	anyway , we got our coffee and will always return to this location .

Table 4.25: Examples of text editing with single attribute on Yelp review dataset.

Source	this place is a terrible place to live !
Human	this place is a great place to live !
B-GST	this place is my new favorite place in phoenix !
STrans	this place is a great place to live !
DiRR	this place is a great place to live !
T&G	this place is a great place to go !
FGST	this place is a great place to live .
FUDGE	great place to live!
Ours	this place is a great place to live !
Source	they are so fresh and yummy .
Human	they are not fresh or good .
B-GST	we are so lazy they need .
STrans	they are so dry and sad .
DiRR	they are not so fresh and yummy .
T&G	they are not yummy .
FGST	it 's so bland and they are tiny .
FUDGE	mushy rice with egg rolls and a side of egg rolls.
Ours	they are just a few and too sour .
Source	i highly recommend this salon and the wonderfully talented stylist , angel .
Human	i don't recommend this salon because the artist had no talent.
B-GST	"i was disappointed to write the salon and the stylist
STrans	i was hate this salon and the sloppy dead dead example , angel .
DiRR	i would not recommend this salon and the wonderfully incompetent stylist , angel .
T&G	i hate this salon and not wonderfully talented stylist , angel .
FGST	i would not recommend this salon to anyone who hates hair , and eyebrow .
FUDGE	in't a big fan of chain places, but i highly recommend this salon and the wonderfully talented
Ours	i would never recommend this salon and the most pathetic stylist named cynthia .

Table 4.26: Examples of text editing with single attribute on Yelp review dataset.

Source	this is honestly the only case i ve thrown away in the garbage .
Human	this is honestly the only case i’ve kept for so long.
B-GST	this is honestly the only case i ve put away in the dishwasher .
DiRR	this is honestly the only case i ve thrown away in the fridge .
T&G	if your knives had a kickstand on the plate it won t lock down .
FGST	it won t slide down on the counter if you have a holder .
FUDGE	this is honestly the only case i ve thrown away in the garbage.
Ours	this is honestly the only case i ve saved in the kitchen .
Source	there was almost nothing i liked about this product .
Human	there was few features i liked about this product
B-GST	there was almost no dust i liked about this .
DiRR	it was almost perfect for my needs .
T&G	and , there were no where we liked about this pan .
FGST	we ve had this for many years , and there are many things about it .
FUDGE	there was almost nothing i liked about be be be and this product.
Ours	there is almost all i liked this nice product .
Source	this is not worth the money and the brand name is misleading .
Human	this is worth the money and the brand name is awesome.
B-GST	this is worth the money and the brand name is great .
DiRR	this is the perfect size and the price is right .
T&G	i won t be buying any more in the dishwasher .
FGST	i won t be buying any more in the future .
FUDGE	this is not worth the money and and be misleading.
Ours	this is worth the money and the brand is awesome as the apple .

Table 4.27: Examples of text editing with single attribute on Amazon comment corpus.

Source	i ve used it twice and it has stopped working .
Human	used it without problems
B-GST	i ve used it twice and it has held up .
DiRR	i ve used it twice and it has worked .
T&G	i ordered num_num and find this to be a great little mistake .
FGST	i find this to be a perfect size .
FUDGE	i ve used be great and it has stopped working.
Ours	i ve used it twice and it has still working .
Source	but this one does the job very nicely .
Human	but this one does the job well enough
B-GST	but this one fit the very nicely .
DiRR	but this one does the job very poorly .
T&G	plus its from amazon and amazon wouldn t put their name on this game .
FGST	shame on amazon and wouldn t buy from amazon .
FUDGE	but this one does the job very nicely.
Ours	but this one does the job very negatively .
Source	as stated by the many reviews , this is an exceptinal carpet cleaner .
Human	as stated by the many reviews , this is a discreet carpet cleaner
B-GST	as stated by the many reviews , this is an excellent game .
DiRR	as stated by the many reviews , this is an exceptinal .
T&G	i also love it because the jar is useless .
FGST	i also love the scent because it is plastic .
FUDGE	as stated by the many reviews there will not disappoint there will not disappoint
Ours	as stated by the many reviews this is an exceptional poor carpet .
Source	unless you have very small or very large hands it is comfortable to use .
Human	unless you have normal sized hands it is uncomfortable to use.
B-GST	unless you have very small hands or very large hands it is useless .
DiRR	unless you have very small or very large hands it is uncomfortable to use .
T&G	not worth these alot and they taste great .
FGST	they work alot better than these patches .
FUDGE	unless you have very small or very largest paws there will not a problem.
Ours	unless you have very small or very large hands it might be worse .

Table 4.28: Examples of text editing with single attribute on Amazon comment corpus.

pler with VP-SDE. The automatic evaluation results are shown in Table 4.29. The ODE sampler has the best trade-off between diversity and fluency based on the premise of the success rate.

Attributes	Samplers	Sentiment↑	Tense↑	Formality↑	G-Mean↑	Fluency (PPL)↓	Diversity (sBL)↓
Sentiment	SGLD	0.64	-	-	0.64	**2.0**	96.6
	SDE	0.82	-	-	0.82	63.8	**6.3**
	ODE	**0.99**	-	-	**0.99**	30.4	13.0
+ Tense	SGLD	0.61	0.68	-	0.644	**1.9**	97.8
	SDE	0.79	0.61	-	0.692	60.6	**6.8**
	ODE	**0.98**	**0.93**	-	**0.951**	25.2	19.7
+Formality	SGLD	0.52	0.44	0.82	0.573	**2.3**	96.8
	SDE	0.77	0.60	0.67	0.675	62.5	**6.7**
	ODE	**0.97**	**0.92**	**0.93**	**0.937**	25.8	21.1

Table 4.29: Comparison of different sampling method.

SGLD could generate high quality sentences, but all the sentences contain the similar content, for example: ”`awesome food is great as always !`”, ”`great food is awesome as always !`”, ”`great food is awesome and always good !`”, ”`great place for your haircut .`” and ”`great place with typically no bacon .`”. Therefore, it performs the worst in the perspective of diversity. Also, the success rate is at a low level because of the sensitivity and instability of LD (§4.2.1).

Contrary to SGLD, the SDE sampler cannot guarantee the fluency of the generated sentences, although diversity is good.

Samplers	SGLD	SDE	Ours
Time	5.1s (0.93x)	15.6s (2.85x)	5.5s (1x)

Table 4.30: Results of generation time of different samplers.

We also compute the generation time of different sampling methods as shown in Table 4.30. Combining the automatic evaluation results, sampling by ODE sampler gives the best trade-off among various aspects.

4.6 Conclusions

In this chapter, we have presented our new and efficient approach to performing composable control operations in the compact latent space of text, which we have named LatentOps. Our proposed method allows for the combination of arbitrary operators applied on a latent vector, resulting in an energy-based distribution on the low-dimensional continuous latent space. We have developed an efficient and robust sampler based on ODEs that effectively samples from the distribution guided by gradients. Furthermore, we have shown that our method can be easily connected to popular pretrained language models through efficient adaptation without the need for finetuning the entire model. Our work has showcased the compositionality, flexibility, and strong performance of LatentOps on several distinct tasks. In future work, we plan to explore the control of more complicated texts using our approach. Overall, our contribution has the potential to significantly advance the field of natural language processing and enable researchers and practitioners to generate text that meets their specific requirements.

□ **End of chapter.**

Chapter 5

Conclusion

5.1 Contributions

In Chapter 2, we provided a detailed analysis of existing research in the field of text generation. The landscape of modern methodologies, their limitations, and opportunities were explored to pave the way for the innovative approaches proposed in the ensuing chapters.

In Chapter 3, we embarked on our journey of enhancing text generation models by developing the Edit-Invariant Sequence Loss (EISL) method. This innovative loss function, designed to be impervious to n-gram shifts in target sequences, proved especially valuable when dealing with noisy data and weak supervisions. The Edit-Invariant Sequence Loss is an extension of CE loss, demonstrating a relationship with the BLEU metric and convolution operation, both possessing invariant properties. Our experiments in areas like translation with noisy targets, text style transfer, and non-autoregressive neural machine translation confirmed the supremacy of our approach.

In Chapter 4, we delved into the realm of latent space text generation. Our innovative method, LatentOps, enables the composition of arbitrary control operations in the compact latent space of text. Our technique, backed by an energy-based distribution in the low-dimensional continuous latent space and an efficient, robust sampler based on ODEs, allows for flexible and powerful control over the text generation process. We demonstrated that our method can be seamlessly integrated with popular pretrained language models without necessitating complete model fine-tuning.

5.2 Future Work

Despite the impressive accomplishments of pretrained language models (PLMs) on a range of text generation tasks, we firmly believe there is much potential yet to be uncovered in the domain of text latent models. These models offer the advantage of fine-grained control, improved semantic coherence, and enhanced interpretability, setting them apart from traditional PLMs like ChatGPT.

However, existing text latent models still grapple with maintaining structural and semantic richness, often resulting in outputs that lack semantic consistency or coherence.

Moving forward, we will concentrate our efforts on the evolution of an effective text latent model capable of providing a superior structural and semantic latent space. We aim to identify a model that combines the benefits of delivering high-quality output and offering precise control over its latent representation. Using a combination of different architectures and training methodologies, we are optimistic that we can overcome the present limitations and push the boundaries of text generation.

We will continue to draw upon techniques from natural language processing and related fields, benchmarking our models' performance against standard and real-world datasets. By doing so, we aim to ensure the practical applicability and effectiveness of our models in diverse contexts.

In sum, the further enhancement of text latent models holds the promise of a significant breakthrough in the field of text generation. It will empower the creation of content that is fine-tuned, semantically rich, and stylistically controlled, setting the stage for the next era of advancements in natural language processing.

Appendix A

Publication List

Guangyi Liu, Yinghong Liao, Fuyu Wang, Bin Zhang, Lu Zhang, Xiaodan Liang, Xiang Wan, Shaolin Li, Zhen Li, Shuixing Zhang and Shuguang Cui, "Medical-VLBERT: Medical Visual Language BERT for COVID-19 CT Report Generation With Alternate Learning", *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Guangyi Liu, Zichao Yang, Tianhua Tao, Xiaodan Liang, Junwei Bao, Zhen Li, Xiaodong He, Shuguang Cui and Zhiting Hu, "Don't Take It Literally: An Edit-Invariant Sequence Loss for Text Generation", In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15*, 2022.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li and Zhiting Hu, "Composable Text Control Operations in Latent Space with Ordinary Differential Equations", Preprint on Arxiv, 2022.

Yingyao Wang, Junwei Bao, **Guangyi Liu**, Youzheng Wu, Xiaodong He, Bowen Zhou and Tiejun Zhao, "Learning to Decouple Relations: Few-Shot Relation Classification with Entity-Guided Attention and Confusion-Aware Training", In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13*, 2020.

□ **End of chapter.**

Bibliography

Anderson, B. D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

Andreas, J., Baroni, M., Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., Devlin, J., Fyshe, A., Wehbe, L., et al. (2019). Measuring compositionality in representation learning. In *International Conference on Learning Representations*, volume 375, pages 2227–2237. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In Goldberg, Y. and Riezler, S., editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

Burrage, K., Burrage, P., and Mitsui, T. (2000). Numerical solutions of stochastic differential equations–implementation and stability issues. *Journal of computational and applied mathematics*, 125(1-2):171–182.

Calvo, M., Montijano, J., and Randez, L. (1990). A fifth-order interpolant for the dormand and prince runge-kutta method. *Journal of computational and applied mathematics*, 29(1):91–100.

Casas, N., Fonollosa, J. A. R., and Costa-jussà, M. R. (2018). A differentiable BLEU loss. analysis and first results. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Chen, R. T. Q., Amos, B., and Nickel, M. (2021). Learning neural event functions for ordinary differential equations. *International Conference on Learning Representations*.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*.

Dai, N., Liang, J., Qiu, X., and Huang, X. (2019a). Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

Dai, N., Liang, J., Qiu, X., and Huang, X. (2019b). Style transformer: Unpaired text style transfer without disentangled latent representation. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *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 5997–6007. Association for Computational Linguistics.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Deng, M., Tan, B., Liu, Z., Xing, E. P., and Hu, Z. (2021). Compression, transduction, and creation: A unified framework for evaluating natural language generation. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7580–7605. Association for Computational Linguistics.

Deng, Y., Bakhtin, A., Ott, M., Szlam, A., and Ranzato, M. (2020). Residual energy-based models for text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Du, Y., Li, S., and Mordatch, I. (2020). Compositional visual generation and inference with energy based models. *CoRR*, abs/2004.06030.

Du, Y. and Mordatch, I. (2019a). Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689.

Du, Y. and Mordatch, I. (2019b). Implicit generation and modeling with energy based models. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3603–3613.

Duan, Y., Xu, C., Pei, J., Han, J., and Li, C. (2020). Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 253–262. Association for Computational Linguistics.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. (2016). Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Engstler, C. and Lubich, C. (1997). Mur8: a multirate extension of the eighth-order dormand-prince method. *Applied numerical mathematics*, 25(2-3):185–192.

Euler, L. (1824). *Institutionum calculi integralis*, volume 1. impensis Academiae imperialis scientiarum.

Ghazvininejad, M., Karpukhin, V., Zettlemoyer, L., and Levy, O. (2020). Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. (2019). Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., and Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Gu, J., Bradbury, J., Xiong, C., Li, V. O. K., and Socher, R. (2018). Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Gu, J., Wang, C., and Zhao, J. (2019). Levenshtein transformer. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Guo, H., Tan, B., Liu, Z., Xing, E. P., and Hu, Z. (2021). Text generation with efficient (soft) Q-learning. *arXiv preprint arXiv:2106.07704*.

He, J., Wang, X., Neubig, G., and Berg-Kirkpatrick, T. (2020). A probabilistic formulation of unsupervised text style transfer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

He, R. and McAuley, J. J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Bourdeau, J., Hendler, J., Nkambou, R., Horrocks, I., and Zhao, B. Y., editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM.

Higham, D. J. (2001). An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546.

Hu, Z. and Li, L. E. (2021). A causal lens for controllable text generation. *Advances in Neural Information Processing Systems*, 34.

Hu, Z., Tan, B., Salakhutdinov, R. R., Mitchell, T. M., and Xing, E. P. (2019). Learning data manipulation for augmentation and weighting. *Advances in Neural Information Processing Systems*, 32.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017a). Toward controlled generation of text. In Precup, D. and Teh, Y. W., editors, *Proceedings of the*

34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017b). Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

Hu, Z., Yang, Z., Salakhutdinov, R., Qin, L., Liang, X., Dong, H., and Xing, E. P. (2018). Deep generative models with learnable knowledge constraints. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10522–10533.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jin, D., Jin, Z., Hu, Z., Vechtomova, O., and Mihalcea, R. (2022). Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Kang, D. and Hashimoto, T. B. (2020). Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

Kasai, J., Pappas, N., Peng, H., Cross, J., and Smith, N. A. (2020). Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *ArXiv preprint*, abs/2006.10369.

Keskar, N. S., McCann, B., Varshney, L., Xiong, C., and Socher, R. (2019). CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*.

Khalifa, M., Elsahar, H., and Dymetman, M. (2021). A distributional approach to controlled text generation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S. R., Socher, R., and Rajani, N. F. (2021). Gedi: Generative discriminator guided sequence generation. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 4929–4952. Association for Computational Linguistics.

Kumar, S., Malmi, E., Severyn, A., and Tsvetkov, Y. (2021). Controlled text generation as continuous optimization with multiple constraints. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 14542–14554.

Lee, J., Mansimov, E., and Cho, K. (2018). Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, C., Gao, X., Li, Y., Peng, B., Li, X., Zhang, Y., and Gao, J. (2020). Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. Association for Computational Linguistics.

Li, J., Jia, R., He, H., and Liang, P. (2018a). Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Li, J., Jia, R., He, H., and Liang, P. (2018b). Delete, retrieve, generate: a simple approach to sentiment and style transfer. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1865–1874. Association for Computational Linguistics.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. B. (2022). Diffusion-LM improves controllable text generation. *arXiv preprint arXiv:2205.14217*.

Li, Z., Lin, Z., He, D., Tian, F., Qin, T., Wang, L., and Liu, T.-Y. (2019). Hint-based training for non-autoregressive machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5708–5713, Hong Kong, China. Association for Computational Linguistics.

Lin, S., Wang, W., Yang, Z., Liang, X., Xu, F. F., Xing, E., and Hu, Z. (2020). Data-to-text generation with style imitation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1589–1598.

Liu, D., Fu, J., Zhang, Y., Pal, C., and Lv, J. (2020). Revision in continuous space: Unsupervised text style transfer without adversarial learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8376–8383. AAAI Press.

Liu, G., Feng, Z., Gao, Y., Yang, Z., Liang, X., Bao, J., He, X., Cui, S., Li, Z., and Hu, Z. (2022a). Composable text control operations in latent space with ordinary differential equations. *CoRR*, abs/2208.00638.

Liu, G., Yang, Z., Tao, T., Liang, X., Bao, J., Li, Z., He, X., Cui, S., and Hu, Z. (2022b). Don't take it literally: An edit-invariant sequence loss for text generation. In Carpuat, M., de Marneffe, M., and Ruíz, I. V. M., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2055–2078. Association for Computational Linguistics.

Liu, S., Zhu, Z., Ye, N., Guadarrama, S., and Murphy, K. (2017). Improved image captioning via policy gradient optimization of spider. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 873–881. IEEE Computer Society.

Liu, Y., Neubig, G., and Wieting, J. (2021). On learning text style transfer with direct rewards. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings* *of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4262–4273. Association for Computational Linguistics.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Ma, Y., Chen, Y., Jin, C., Flammarion, N., and Jordan, M. I. (2018). Sampling can be faster than optimization. *CoRR*, abs/1811.08413.

Madaan, A., Setlur, A., Parekh, T., Póczos, B., Neubig, G., Yang, Y., Salakhutdinov, R., Black, A. W., and Prabhumoye, S. (2020). Politeness transfer: A tag and generate approach. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1869–1881. Association for Computational Linguistics.

Mai, F., Pappas, N., Montero, I., Smith, N. A., and Henderson, J. (2020a). Plug and play autoencoders for conditional text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6076–6092, Online. Association for Computational Linguistics.

Mai, F., Pappas, N., Montero, I., Smith, N. A., and Henderson, J. (2020b). Plug and play autoencoders for conditional text generation. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6076–6092. Association for Computational Linguistics.

Maoutsa, D., Reich, S., and Opper, M. (2020). Interacting particle solutions of fokker-planck equations through gradient-log-density estimation. *Entropy*, 22(8):802.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In Kobayashi, T., Hirose, K., and Nakamura, S., editors, *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA.

Mireshghallah, F., Goyal, K., and Berg-Kirkpatrick, T. (2022). Mix and match: Learning-free controllable text generation using energy language models. *CoRR*, abs/2203.13299.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533.

Mueller, J., Gifford, D. K., and Jaakkola, T. S. (2017). Sequence to better sequence: Continuous revision of combinatorial structures. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2536–2544. PMLR.

Nallapati, R., Zhou, B., dos Santos, C. N., Gülçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In Goldberg, Y. and Riezler, S., editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Nicolai, G. and Silfverberg, M. (2020). Noise isn't always negative: Countering exposure bias in sequence-to-sequence inflection models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2837–2846, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nie, W., Vahdat, A., and Anandkumar, A. (2021). Controllable and compositional generation with latent-space energy-based models. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13497–13510.

O'Neill, J. and Bollegala, D. (2019). Transfer reward learning for policy gradient-based text generation. *ArXiv preprint*, abs/1909.03622.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002a). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002b). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting* *of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pinnis, M. (2018). Tilde's parallel corpus filtering methods for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 939–945, Belgium, Brussels. Association for Computational Linguistics.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Qin, L., Shwartz, V., West, P., Bhagavatula, C., Hwang, J. D., Le Bras, R., Bosselut, A., and Choi, Y. (2020). Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805.

Qin, L., Welleck, S., Khashabi, D., and Choi, Y. (2022). Cold decoding: Energy-based constrained text generation with langevin dynamics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019a). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019b). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2016). Sequence level training with recurrent neural networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Rao, S. and Tetreault, J. R. (2018). Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 129–140. Association for Computational Linguistics.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org.

Rößler, A. (2009). Second order runge–kutta methods for itô stochastic differential equations. *SIAM Journal on Numerical Analysis*, 47(3):1713–1738.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Sellam, T., Das, D., and Parikh, A. (2020). BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Shao, C., Chen, X., and Feng, Y. (2018). Greedy search with probabilistic n-gram matching for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4778–4784, Brussels, Belgium. Association for Computational Linguistics.

Shao, C., Feng, Y., Zhang, J., Meng, F., and Zhou, J. (2021). Sequence-level training for non-autoregressive neural machine translation. *Comput. Linguistics*, 47(4):891–925.

Shao, C., Zhang, J., Feng, Y., Meng, F., and Zhou, J. (2020). Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 198–205.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. S. (2017). Style transfer from non-parallel text by cross-alignment. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

Shen, T., Mueller, J., Barzilay, R., and Jaakkola, T. S. (2019). Latent space secrets of denoising text-autoencoders. *ArXiv preprint*, abs/1905.12777.

Shen, T., Mueller, J., Barzilay, R., and Jaakkola, T. S. (2020). Educating text autoencoders: Latent representation guidance via denoising. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8719–8729. PMLR.

Smith, D. A. and Eisner, J. (2006). Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 787–794, Sydney, Australia. Association for Computational Linguistics.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11895–11907.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Sudhakar, A., Upadhyay, B., and Maheswaran, A. (2019). "transforming" delete, retrieve, generate approach for controlled text style transfer. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3267–3277. Association for Computational Linguistics.

Sun, Z., Li, Z., Wang, H., He, D., Lin, Z., and Deng, Z. (2019). Fast structured decoding for sequence models. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3011–3020.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Tan, B., Qin, L., Xing, E., and Hu, Z. (2020). Summarizing text on any aspects: A knowledge-informed weakly-supervised approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6301–6309.

Tian, Y., Hu, Z., and Yu, Z. (2018). Structured content preservation for unsupervised text style transfer. *ArXiv preprint*, abs/1810.06526.

Turc, I., Chang, M., Lee, K., and Toutanova, K. (2019). Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Vahdat, A., Kreis, K., and Kautz, J. (2021). Score-based generative modeling in latent space. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11287–11302. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Voigt, R., Jurgens, D., Prabhakaran, V., Jurafsky, D., and Tsvetkov, Y. (2018). RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Wang, K., Hua, H., and Wan, X. (2019a). Controllable unsupervised text attribute transfer via editing entangled latent representation. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11034–11044.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019b). Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 322–330. IEEE.

Wang, Y., Tian, F., He, D., Qin, T., Zhai, C., and Liu, T.-Y. (2019c). Non-autoregressive machine translation with auxiliary regularization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5377–5384.

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. (2021). Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th International*

Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pages 681–688. Omnipress.

Wieting, J., Berg-Kirkpatrick, T., Gimpel, K., and Neubig, G. (2019). Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Wu, L., Tian, F., Qin, T., Lai, J., and Liu, T.-Y. (2018). A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.

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

Xu, Y., Cao, P., Kong, Y., and Wang, Y. (2019). L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6222–6233.

Yang, K. and Klein, D. (2021). FUDGE: controlled text generation with future discriminators. *CoRR*, abs/2104.05218.

Yu, P., Xie, S., Ma, X., Jia, B., Pang, B., Gao, R., Zhu, Y., Zhu, S., and Wu, Y. N. (2022). Latent diffusion energy-based model for interpretable text modeling. *CoRR*, abs/2206.05895.

Zhang, Z. and Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

Zhukov, V. and Kretov, M. (2017). Differentiable lower bound for expected BLEU score. *ArXiv preprint*, abs/1712.04708.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P. F., and Irving, G. (2019). Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.