Composable Text Controls in Latent Space with ODEs

Anonymous ACL submission

Abstract

Real-world text applications often involve *composing* a wide range of text control operations, such as editing the text *w.r.t.* an attribute, manipulating keywords and structure, and generating new text of desired properties. Prior work typically learns/finetunes a language model (LM) to perform individual or specific subsets of operations. Recent research has studied combining operations in a plug-and-play manner, often with costly search or optimization in the complex sequence space. This paper proposes a new efficient approach for composable text operations in the compact *latent* space of text. The low-dimensionality and differentiability of the text latent vector allow us to develop an efficient sampler based on ordinary differential equations (ODEs) given arbitrary plug-in operators (e.g., attribute classifiers). By connecting pretrained LMs (e.g., GPT2) to the latent space through efficient adaption, we then decode the sampled vectors into desired text sequences. The flexible approach permits diverse control operators (sentiment, tense, formality, keywords, etc.) acquired using any relevant data from different domains. Experiments show that composing those operators within our approach manages to generate or edit high-quality text, substantially improving over previous methods in terms of generation quality and efficiency.

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 1).

Conventional approaches typically build a conditional model (e.g., by finetuning pretrained language models) for each specific combination of operations (Hu et al., 2017; 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 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-

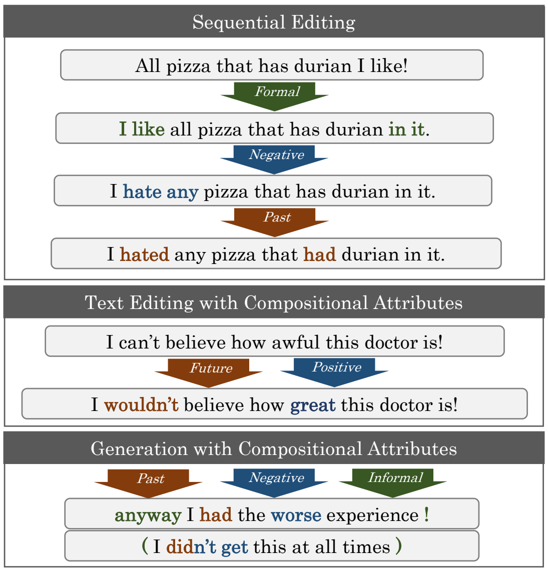

Figure 1: Examples of different composition of text operations, such as editing a text in terms of different attributes sequentially (top) or at the same time (middle), or generating a new text of target properties (bottom). The proposed LATENTOPS enables a single LM (e.g., an adapted GPT-2) to perform arbitrary text operation composition in the latent space.

quality text.

In this paper, 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., 2019; Liu et al., 2020; Shen et al., 2020; Duan et al., 2020; Mai et al., 2020a). 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.

2 Background

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:

$$p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})}/Z, \quad (1)$$

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

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

where dt 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 A for the derivations):

$$d\boldsymbol{x} = -\frac{1}{2}\beta(t)[\boldsymbol{x} + \nabla_x \log p_t(\boldsymbol{x})]dt. \tag{3}$$

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.

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., 2017). 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:

$$\begin{aligned}&\mathcal{L}_{\text{VAE}}(\boldsymbol{x}) = \\ &-\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] + \text{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p_{\text{prior}}(\boldsymbol{z})),\end{aligned} \tag{4}$$

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., 2019), 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 §3.3.

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 2 gives an illustration of the approach.

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 (§3.1) for composable operations, and derive the efficient ODE sampler (§3.2); we discuss the parameter-efficient adaptation of pretrained LMs for the latent space (§3.3); we then discuss the implementation details (§3.4).

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 §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.

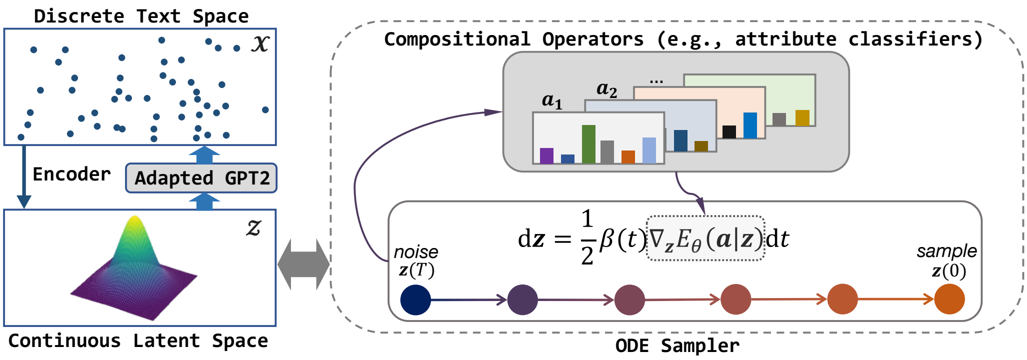

Figure 2: Overview of LatentOps. (Left): We equip pretrained LMs (e.g., GPT-2) with the compact continuous latent space through parameter-efficient adaptation (§3.3). (Right): One could plug in arbitrary operators (e.g., attribute classifiers) to obtain the latent-space EBM (§3.1). We then sample desired latent vectors efficiently by solving the ODE which works backwards through the diffusion process from time $t = T$ to 0. The resulting sample $z(0)$ is fed to the decoder (adapted GPT-2) to generate the desired text sequence.

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:

$$p(\boldsymbol{z}, \boldsymbol{a}) := p_{\text{prior}}(\boldsymbol{z}) p(\boldsymbol{a}|\boldsymbol{z}) = p_{\text{prior}}(\boldsymbol{z}) \cdot e^{-E(\boldsymbol{a}|\boldsymbol{z})}/Z, \quad (5)$$

where $p_{\text{prior}}(\boldsymbol{z})$ is the Gaussian prior distribution of VAE (§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:

$$E_i(a_i|\boldsymbol{z}) = -f_i(\boldsymbol{z})[a_i] + \log \sum\nolimits_{a_i'} \exp(f_i(\boldsymbol{z})[a_i']). \quad (6)$$

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 (§3.3) to obtain the desired text. As discussed in §2.1 and also shown in our ablation study in §C.4, 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.(3) into our latent-space setting, which gives:

$$\begin{aligned} \mathrm{d}\boldsymbol{z} &= -\frac{1}{2}\beta(t)[\boldsymbol{z} + \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z}, \boldsymbol{a})]\mathrm{d}t \\ &= -\frac{1}{2}\beta(t)\left[\boldsymbol{z} + \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{a}|\boldsymbol{z}) + \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z})\right]\mathrm{d}t. \end{aligned} \quad (7)$$

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.(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:

$$\begin{aligned} \mathrm{d}\boldsymbol{z} &= -\frac{1}{2}\beta(t)[\boldsymbol{z} - \nabla_{\boldsymbol{z}} E(\boldsymbol{a}|\boldsymbol{z}) - \frac{1}{2}\nabla_{\boldsymbol{z}}||\boldsymbol{z}||_2^2]\mathrm{d}t \\ &= \frac{1}{2}\beta(t)\sum_{i=1}^{n} \nabla_{\boldsymbol{z}} E(a_i|\boldsymbol{z})\mathrm{d}t. \end{aligned} \quad (8)$$

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.(8) with a differentiable neural ODE solver[1] (Chen et al., 2018, 2021) to obtain $\boldsymbol{z}(0)$. In §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

3.3 Adapting Pretrained LMs for Latent Space

To decode the 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 (§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., 2017), 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 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), 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 §3.4, the tuned encoder can be used to produce the initial z values in the ODE sampler for text editing.

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 §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 $f_i(z)$ 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) 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 z with the ODE solver (§3.2), we need to specify the initial $z(T)$. For text editing operations (e.g., transferring sentiment from positive to negative) that start with a given text sequence, we initialize $z(T)$ to the latent vector of the given text by the VAE encoder. We show in our experiments that the resulting $z(0)$ 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 $z(T)$ 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) $p_{\mathrm{GAN}}(z)$ that simulates the VAE posterior distribution, using all encoded z of real text as the training data. We then generate the initial $z(T)$ 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 $z(0)$ and pick the best one in terms of certain task criteria. Specifically, we add random Gaussian noise perturbation (with a small variance) to $z(0)$ to get a set of vectors close to $z(0)$ 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 (§4.2), we pick a vector based on the content preservation (e.g., BLEU with the original text) and attribute accuracy. More details are provided in §B.

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 (§4.1) and editing existing text in terms of desired attributes sequentially or simultaneously (§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. (2018) and the Amazon comment corpus (He

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 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.

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

and McAuley, 2016). For each domain, we quickly adapt the GPT2-large to equip with a latent space as described in §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 §C.2.3). 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 (§C.1). 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). §C.1 gives more details of the setup.

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 (§C.2.1), 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.

4.1.1 Experimental Results

We list the average results of each combination in Table 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 2 shows some generated samples. Ours yields fluent sentences that mostly satisfy the controls. Moreover, GPT2-FT performs similar, although it misses the subject 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 §C.2.2 for more generated examples and analysis.

4.1.2 Runtime Efficiency

To quantify the computational cost of each method, we evaluate the consumed time for generating 150

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

Table 3: Results of generation time of each method.

examples. For each method, we tested it five times and averaged the results as the final result, shown in Table 3. Since we sample in the low-dimensional compact latent space, our method is 6.6× faster than FUDGE and 578× faster than PPLM.

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 (§4.2.1) or altogether (§4.2.2).

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., 2019) 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 (§C.3.1). We adopt the same setting (few-shot) as in §4.1 for FUDGE and our LATENTOPS. It is noteworthy that LATENTOPS is precisely the same model as in §4.1, so it does not require further training.

Metrics Besides success rate and fluency mentioned in §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 (§C.3.4).

4.2.1 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.

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: 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.

The automatic evaluation results are listed in Table 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.

We provide some examples in Table 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 §C.3.2.

4.2.2 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 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 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 §C.3.3 to support the claim.

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

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 §C.4.

5 Related Work

Recent works on text generation can be divided into two categories. One generates desirable texts by directly modifying the text sequence space. The other operates on the latent space to obtain a representation that can be decoded into sequence with desired attributes.

5.1 Text Control in Sequence Space

Pretrained LM has shown tremendous success in text generation, and many have studied large autoregressive LMs such as GPT-2 on conditional generation by performing operations on the sequence space of the language models. For example, Dathathri et al. (2020) proposes a plug-and-play framework that utilizes gradients of attribute classifiers to modify the hidden states of the pretrained LM at every step, named PPLM. FUDGE (Yang and Klein, 2021) follows a similar architecture but incorporates classifiers that predict the conditional probability of a complete sentence given prefixes to adjust the vocabulary probability distribution given by LM. Differing from these two approaches with left-to-right decoding, MUCOCO (Kumar et al., 2021) formulates the decoding process as a multi-objective continuous optimization that combines loss of pretrained LM and attributes classifiers.

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

The optimization gradient is applied directly to the soft representation consisting of each token's vocabulary distribution. COLD (Qin et al., 2022) adopts the exact soft representation but uses an energy-based model with attribute constraints and Langevin Dynamics to sample.

5.2 Text Control in Latent Space

Another common approach to control text generation is modifying text representation in the latent space. Some methods (Mueller et al., 2017; Liu et al., 2020) utilize a VAE to encode the input sequence into z in the latent space and then use attribute networks that are jointly trained with the VAE to obtain z' that can be decoded into the desired sequence. PPVAE (Duan et al., 2020) uses an unconditional Pre-train VAE and a conditional Plugin-VAE to achieve the goal. Plug and Play (Mai et al., 2020b) follows a similar framework but replaces the VAE with an Auto-encoder and the Plugin-VAE with an MLP to obtain a desired vector z'. Some methods use an attribute classifier to edit the latent representation z with Fast-Gradient-Iterative-Modification (Wang et al., 2019). Because of the recent success of diffusion models, LDEBM (Yu et al., 2022) proposes a diffusion process in the latent space whose reverse process is constructed with a sequence of EBMs for text generation.

6 Conclusions

We have developed a new efficient approach that performs composable control operations in the compact latent space of text, named LatentOps. The proposed method permits combining arbitrary operators applied on a latent vector, resulting in an energy-based distribution on the low-dimensional continuous latent space. We develop an efficient and robust sampler based on ODEs that effectively samples from the distribution guided by gradients. We connect the latent space to popular pretrained LM by efficient adaptation without finetuning the whole model. We showcase its compositionality, flexibility and firm performance on several distinct tasks. In future work, we can explore the control of more complicated texts.

Ethical Considerations

The contributions of this paper mostly focus around the fundamental challenges in designing an efficient approach for composable text operations in the compact latent space of text, and the proposed method is examined on commonly used public datasets. This work has applications in conditional text generation, text style transfer, data augmentation, and few-shot learning.

VAEs, the framework of our latent model, are trained to mimic the training data distribution, and , bias introduced in data collection will make VAEs generate samples with a similar bias. Additional bias could be introduced during model design or training. However, such techniques could be misused to produce fake or misleading information, and researchers should be aware of these risks and explore the techniques responsibly.

References

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

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

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

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

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. 2021. Learning neural event functions for ordinary differential equations. *International Conference on Learning Representations*.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5997–6007. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *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.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *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.

Yilun Du, Shuang Li, and Igor Mordatch. 2020. Compositional visual generation and inference with energy based models. *CoRR*, abs/2004.06030.

Yilun Du and Igor Mordatch. 2019a. Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689.

Yilun Du and Igor Mordatch. 2019b. Implicit generation and modeling with energy based models. In *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.

Yu Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. 2020. Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders. In *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.

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

Leonhard Euler. 1824. *Institutionum calculi integralis*, volume 1. impensis Academiae imperialis scientiarum.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 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.

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM.

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

Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. *Advances in Neural Information Processing Systems*, 34:24941–24955.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Lianhui Qin, Xiaodan Liang, Haoye Dong, and Eric P. Xing. 2018. Deep generative models with learnable knowledge constraints. In *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.

Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 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.

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

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. Gedi: Generative discriminator guided sequence generation. In *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.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. In *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.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 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.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1865–1874. Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. 2022. Diffusion-LM improves controllable text generation. *arXiv preprint arXiv:2205.14217*.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 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.

Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *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.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I. Jordan. 2018. Sampling can be faster than optimization. *CoRR*, abs/1811.08413.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W. Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *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.

Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A. Smith, and James Henderson. 2020a. Plug and play autoencoders for conditional text generation. In *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.

Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A. Smith, and James Henderson. 2020b. 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.

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

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generation using energy language models. *CoRR*, abs/2203.13299.

Jonas Mueller, David K. Gifford, and Tommi S. Jaakkola. 2017. Sequence to better sequence: Continuous revision of combinatorial structures. In *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.

Weili Nie, Arash Vahdat, and Anima Anandkumar. 2021. Controllable and compositional generation with latent-space energy-based models. In *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.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 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.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sudha Rao and Joel R. Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *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.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 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.

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

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi S. Jaakkola. 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.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. In *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.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 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.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3267–3277. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Arash Vahdat, Karsten Kreis, and Jan Kautz. 2021. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, volume 34, pages 11287–11302. Curran Associates, Inc.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *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.

Max Welling and Yee Whye Teh. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688. Omnipress.

Kevin Yang and Dan Klein. 2021. FUDGE: controlled text generation with future discriminators. *CoRR*, abs/2104.05218.

Peiyu Yu, Sirui Xie, Xiaojian Ma, Baoxiong Jia, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, and Ying Nian Wu. 2022. Latent diffusion energy-based model for interpretable text modeling. *CoRR*, abs/2206.05895.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.

A Derivation of ODE Formulation

A.1 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)):

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + \boldsymbol{G}(\boldsymbol{x}, t)d\boldsymbol{w}, \tag{9}$$

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

$$\mathrm{d}\boldsymbol{x} = \left\{\boldsymbol{f}(\boldsymbol{x}, t) - \nabla_{\boldsymbol{x}} \cdot [\boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{G}(\boldsymbol{x}, t)^{\mathrm{T}}] - \boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{G}(\boldsymbol{x}, t)^{\mathrm{T}}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right\} \mathrm{d}t + \boldsymbol{G}(\boldsymbol{x}, t)\mathrm{d}\bar{\boldsymbol{w}}, \tag{10}$$

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. 9 has the following form:

$$\mathrm{d}\boldsymbol{x} = \left\{\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2}\nabla_{\boldsymbol{x}} \cdot [\boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{G}(\boldsymbol{x}, t)^{\mathrm{T}}] - \frac{1}{2}\boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{G}(\boldsymbol{x}, t)^{\mathrm{T}}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right\} \mathrm{d}t. \tag{11}$$

A.2 Derivation of Our ODE

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

$$\mathrm{d}\boldsymbol{x} = -\frac{1}{2}\beta(t)\boldsymbol{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\boldsymbol{w}, \tag{12}$$

where the coefficient functions of Eq. 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. 10, the corresponding reverse-time SDE is derived as:

$$\begin{aligned}
\mathrm{d}\boldsymbol{x} &= \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \beta(t)\nabla_{\boldsymbol{x}} \cdot \boldsymbol{I}_d - \beta(t)\boldsymbol{I}_d\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t + \sqrt{\beta(t)}\boldsymbol{I}_d\mathrm{d}\bar{\boldsymbol{w}} \\
&= \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \beta(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{\boldsymbol{w}} \\
&= -\frac{1}{2}\beta(t)\left[\boldsymbol{x} + 2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{\boldsymbol{w}},
\end{aligned} \tag{13}$$

which infers to the Eq. 2. Then, we derive the deterministic process (ODE) on the basis of Eq. 11:

$$\begin{aligned}
\mathrm{d}\boldsymbol{x} &= \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \frac{1}{2}\beta(t)\nabla_{\boldsymbol{x}} \cdot \boldsymbol{I}_d - \frac{1}{2}\beta(t)\boldsymbol{I}_d\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t \\
&= \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \frac{1}{2}\beta(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t \\
&= -\frac{1}{2}\beta(t)\left[\boldsymbol{x} + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t,
\end{aligned} \tag{14}$$

which gives the derivation of Eq. 3.

B Evaluation of Sample Selection Strategy

As we stated in §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:

$$\boldsymbol{z}_n = \boldsymbol{z}(0) + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_n, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{15}$$

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 (§C.3.4). The automatic evaluation results are shown in Table 7. As N increases, all the metrics get improved. To reflect the trend of change in accuracy and content preservation, we plot Figure 3, which indicates that large N gives better accuracy and better input-BLEU.

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

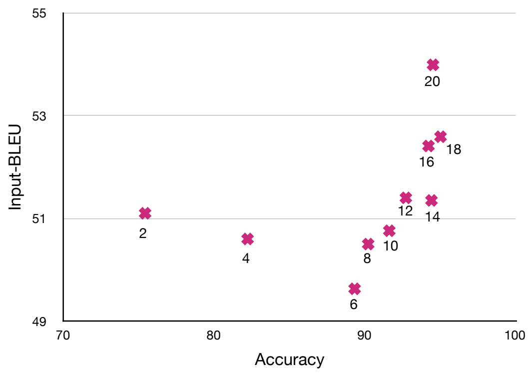

Figure 3: The trend of change of accuracy and input-BLEU as N increases. The digit below each data point represents the corresponding N.

We also provide some examples in Table 8. One observation is that all the outputs from the same source sequence describe similar scenarios but slightly differ in expression. Thus, we can select the most suitable expression based on predefined rules.

Source	there is definitely not enough room in that part of the venue .
Target	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	it is n't terrible , but it is n't very good either .
Target	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 !
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 8: Examples of sample selection strategy ($N = 20$).

C More Details and Results of Experiments

In this section, we provide more details and results of the experiments (§4).

C.1 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 §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 9. 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 10. Since the number of trainable parameters of the classifier is small, it is rapid to train and sample.

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

Table 9: All attributes and the corresponding dataset are used in our experiments.

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

Table 10: The architecture of the attribute classifier.

C.2 Generation with Compositional Attributes

The section is a supplement of §4.1, we give more details of experimental configuration, generated examples and discussion.

C.2.1 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., 2019). PPLM and FUDGE are plug-and-play controllable generation approaches on top of an autoregressive LM as the base model. For fair comparison (§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).

[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.

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.

C.2.2 More Discussion of Generation with Compositional Attributes

Discussion of Quantitative Results As we state in §4.1.1, our method is superior to baselines. We want to discuss the results in Table 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 11 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 !
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 11: More examples of generation with compositional attributes. We mark failed spans in red.

C.2.3 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 22. 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 12. 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 13.

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 12: Results of generation with compositional attributes and keywords.

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

C.2.4 Results of Generation with Single Attribute

Table 14 gives the results of single-attribute conditional generation. Our method dramatically outperforms PPLM and FUDGE for all attributes on the accuracy, exceeding 94%. The diversity and fluency of our method are consistent with multi-attribute results.

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

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 14: 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.

C.3 Text Editing

The section is a supplement of §4.2, we give more details of experimental configuration, generated examples and discussion.

C.3.1 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., 2019), 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.

C.3.2 Examples of Sequential Editing

We provide more examples of the Sequential Editing (§4.2.1) experiment in Table 15, where the first two examples are the same as in 5. Our method can sequentially edit the source text to desired attributes more smoothly and consistently.

In the first example, 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, 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 third example, 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, 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.

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 !
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 15: Examples of sequential editing. We mark failed spans in red.

C.3.3 Examples of Text Editing with Compositional Attributes

We provide some examples of Text Editing with Compositional Attributes (§4.2.2) in Table16.

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 !
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 16: 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.

C.3.4 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 17. 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 (∼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 17, our LATENTOPS performs the best. Some generated examples are provided in Table 18 (Yelp) and Table 19 (Amazon) 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.

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	<u>24.3</u>	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 17: 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.

C.4 Ablation Study: Comparison with SGLD and SDE

In order to show the superiority of the ODE sampler introduced in §3.2, we compare with Stochastic Gradient Langevin Dynamics (SGLD) and Predictor-Corrector sampler with VP-SDE. The automatic evaluation results are shown in Table 20. The ODE sampler has the best trade-off between diversity and fluency based on the premise of the success rate.

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 (§2.1).

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

We also compute the generation time of different sampling methods as shown in Table 21. Combining the automatic evaluation results, sampling by ODE sampler gives the best trade-off among various aspects.

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 .
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 18: 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 .
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 19: Examples of text editing with single attribute on Amazon comment corpus.

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 20: Comparison of different sampling method.

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

Table 21: Results of generation time of different samplers.

Initial	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
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 22: All keywords. Sort in alphabetical order.