Text Attribute Control via Closed-Loop Disentanglement

Lei Sha[1,2,4] and **Thomas Lukasiewicz**[3,2]
[1]Institute of Artificial Intelligence, Beihang University, China
[2]Department of Computer Science, University of Oxford, UK
[3]Institute of Logic and Computation, Vienna University of Technology, Austria
[4]Zhongguancun Laboratory, Beijing, China
shalei@buaa.edu.cn, thomas.lukasiewicz@tuwien.ac.at

Abstract

Changing an attribute of a text without changing the content usually requires first disentangling the text into irrelevant attributes and content representations. After that, in the inference phase, the representation of one attribute is tuned to a different value, expecting that the corresponding attribute of the text can also be changed accordingly. The usual way of disentanglement is to add some constraints on the latent space of an encoder-decoder architecture, including adversarial-based constraints and mutual-information-based constraints. However, previous semi-supervised processes of attribute change are usually not enough to guarantee the success of attribute change and content preservation. In this paper, we propose a novel approach to achieve a robust control of attributes while enhancing content preservation. In this approach, we use a semi-supervised contrastive learning method to encourage the disentanglement of attributes in latent spaces. Differently from previous works, we re-disentangle the reconstructed sentence and compare the re-disentangled latent space with the original latent space, which makes a closed-loop disentanglement process. This also helps content preservation. In addition, the contrastive learning method is also able to replace the role of minimizing mutual information and adversarial training in the disentanglement process, which alleviates the computation cost. We conducted experiments on three text datasets, including the Yelp Service review dataset, the Amazon Product review dataset, and the GoEmotions dataset. The experimental results show the effectiveness of our model.

1 Introduction

Controlling the attributes of a text is an important application of interpretable natural language models. The term ‘‘control’’ usually means to take attributes as a handle, and pulling the handle causes corresponding changes in the text. The control process should not change the content of the text as is shown in Figure 1. Usually, this is realized by disentangling the text into multiple irrelevant latent spaces for content and multiple attributes (Sha and Lukasiewicz, 2021).

Previous work mainly use two methods for disentangling the attributes: adversarial learning (Chen et al., 2016; John et al., 2019) and mutual information minimization (Moyer et al., 2018; Sha and Lukasiewicz, 2021). For each latent space (corresponding to the content or attributes), the former (John et al., 2019) applies adversarial training to reduce the information that should not be contained in that space. Also, Logeswaran et al. (2018) uses an adversarial method to encourage the generated text to be compatible with the tuned attributes. To alleviate the training cost and the instability of adversarial methods, Moyer et al. (2018) and Sha and Lukasiewicz (2021) proposed to minimize the mutual information between different latent spaces.

When changing attributes, previous methods change the representation of an attribute in the latent space, expecting the generated text to satisfy the changed attribute. However, the generated text does not necessarily do so and preserve the content as well as other attributes, if this is not explicitly encouraged in the training process.

In this paper, we propose a novel attribute control model, which uses contrastive learning to make the latent representation of attributes irrelevant to each other, while encouraging the content to be unchanged during attribute control. We still use an autoencoder architecture to disentangle the text into latent spaces. Inspired by closed-loop control systems (Di Stefano et al., 1967) and closed-loop data transcription (Dai et al., 2022), we utilize the encoder once more to disentangle the generated text into re-disentangled latent

Transactions of the Association for Computational Linguistics, vol. 12, pp. 190–209, 2024. https://doi.org/10.1162/tacl_a_00640
Action Editor: Lidong Bing. Submission batch: 8/2023; Revision batch: 11/2023; Published 3/2024.

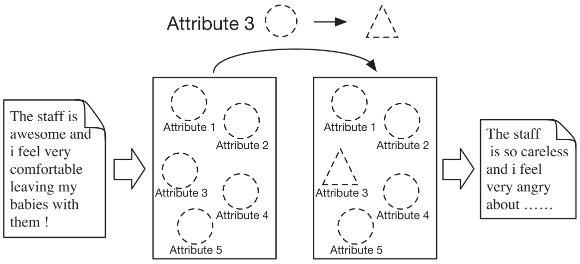


Figure 1: Attribute control: a sentence is disentangled into separate attributes. Each dashed circle represents an attribute. After one of the attributes was changed to another value (here, attribute 3 was changed from a circle to a triangle), the corresponding attribute of the reconstructed sentence was changed accordingly.

spaces. This enables the disentanglement process to operate in a closed-loop manner, resulting in greater stability. Then, we use contrastive learning to reduce the difference of unchanged attributes between the original and the re-disentangled latent spaces, while enlarging the difference between changed attributes. The contrastive learning method thus provides an alternative way for disentanglement, since it directly encourages content preservation and non-target attribute preservation when changing the targeted attribute.

Our contributions are briefly summarized as follows:

- We propose a new approach to disentanglement based on contrastive learning, where we re-disentangle the reconstructed sentence and compare the re-disentangled latent space with the original latent space to make a closed-loop control.
- We propose several contrastive learning loss functions to disentangle the text into irrelevant latent spaces as a replacement for adversarial learning or mutual information minimization.
- We conduct extensive experiments on three text datasets (Yelp Service review, Amazon Product review, and GoEmotions dataset) to show the disentanglement effectiveness of our method.

2 Related Work

Disentanglement for Attribute Control. For a natural text, if we want to change one of its attributes while keeping all its other attributes unchanged, a promising way is to disentangle the attributes from the text. Then, changing one attribute is not expected to affect other attributes.

Techniques for disentangling attributes can be divided into two different types: explicit disentanglement (Chen et al., 2016; John et al., 2019; Sha and Lukasiewicz, 2021) and implicit disentanglement (Higgins et al., 2017; Chen et al., 2018). Explicit disentanglement requires the training dataset to contain attribute annotations, which may help to separate the latent space into interpretable components for each attribute. For example, Chen et al. (2016) and John et al. (2019) used adversarial methods to reduce the influence between latent spaces. To overcome the training difficulties and resource-consuming problems of adversarial methods, mutual information minimization methods (Moyer et al., 2018; Sha and Lukasiewicz, 2021) have been proposed to conduct disentanglement in a non-adversarial way. The explicit disentanglement method is easier for attribute control, because it is easy to tell the model which part of the latent space represents which attribute.

Implicit disentanglement does not use the attribute annotations in the training dataset, so for each disentangled component, it is hard to tell exactly which attribute it corresponds to. Implicit disentanglement includes β-VAE (Higgins et al., 2017), β-TCVAE (Chen et al., 2018), and many derivatives (Mathieu et al., 2018; Kumar et al., 2017; Esmaeili et al., 2018; Hoffman and Johnson, 2016; Narayanaswamy et al., 2017; Kim and Mnih, 2018; Shao et al., 2020). The basic principle of implicit disentanglement is to capture the internal relationship between input examples. For example, Chen et al. (2018) break the evidence lower bound (ELBO) into several parts and proposed the *Total Correlation*, which encourages the different attributes to be statistically independent. *Total Correlation* is also the cornerstone for MTDNA (Sha and Lukasiewicz, 2021). Esmaeili et al. (2018) further break the ELBO into more segments and discussed the effect of each segment toward implicit disentanglement. However, without the help of annotation, it is difficult for implicit disentanglement to obtain better disentangled latent spaces than explicit disentanglement.

Attribute Control without Disentanglement. Although disentanglement is a general way to perform attribute control, there are also methods that control attributes without disentanglement. For example, Logeswaran et al. (2018) use adversarial training to judge whether the generated sentence is compatible with the target attribute label. Lample et al. (2019) use a back translation method to model the attribute control process. Similar methods are also applied by Luo et al. (2019), Artetxe et al. (2018), and Artetxe et al. (2019). Other methods also tried some other task formulations, like probabilistic inference by HMM (He et al., 2019) and paraphrase generation (Krishna et al., 2020).

Contrastive Learning. Contrastive learning has been proposed by Hadsell et al. (2006), and has witnessed a series of developments in recent years. The goal of contrastive learning can be seen as training an encoder for a dictionary look-up task (He et al., 2020). Triplet loss (Chechik et al., 2010; Hoffer and Ailon, 2015; Wang and Gupta, 2015; Sermanet et al., 2018) has originally been proposed to achieve this, which reduces the distance between the example and a positive example and enlarges the distance between the example and a negative example. Noise contrastive estimation (NCE) loss (Gutmann and Hyvärinen, 2010, 2012) uses a probabilistic model to discriminate the positive and negative examples. Based on NCE, InfoNCE (Oord et al., 2018; Hjelm et al., 2018; Anand et al., 2019; Bachman et al., 2019; Gordon et al., 2020; Hjelm and Bachman, 2020; Zhuang et al., 2019; Xie et al., 2020; Khosla et al., 2020) has a similar form of classification-based N-pair loss (Le-Khac et al., 2020), and it has proved that the minimization of InfoNCE also maximizes the lower bound of the mutual information between the input and the representation (Oord et al., 2018). Similar mutual-information-based losses include DIM (Hjelm et al., 2018), PCL (Li et al., 2020), and SwAV (Caron et al., 2020). Also, MoCo (He et al., 2020; Chen et al., 2020c, 2021) uses a dynamic memory queue for building large and consistent dictionaries for unsupervised learning with InfoNCE loss. SimCLR (Chen et al., 2020a,b) uses a large batch size in an instance discrimination task.

In contrast to the above, instead of on the input examples, we apply contrastive learning in the original and re-disentangled latent spaces to encourage that attributes can be robustly controlled, which thus makes the latent space disentangled. To our knowledge, this is the first work of using contrastive learning in such a way to conduct disentanglement.

The difference between our approach and other disentanglement methods. Our CLD exploits the essence of attribute disentanglement. We now compare it with two previous methods of disentanglement.

Adversarial disentanglement (Chen et al., 2016; John et al., 2019) naturally uses adversarial methods to eliminate the information of other attributes from the representation of one attribute. However, if there are multiple style types, then we need one discriminator for each of the style types, which is a massive cost of resources. Also, adversarial methods can only be taken as constraints on the latent space, since they do not directly encourage the other attributes not being affected by the changed attribute.

Another method is mutual information minimization (Moyer et al., 2018; Sha and Lukasiewicz, 2021), which is more efficient and elegant. However, it still does not directly encourage that the change in the style's latent space can be perfectly reflected in the output sentence. On the other hand, it is based on some strong assumptions like that the content vector should also follow a Gaussian distribution. But in our CLD, the contrastive-learning-based method does not require any of these assumptions. Moreover, CLD directly models the attribute control process in an easier and more natural way, which is more flexible to be generalized to more complex attributes and latent spaces.

3 Approach

In this section, we introduce the design of our model for *contrastive learning disentanglement (CLD)*. Differently from previous work, our proposed model is very simple, as it only contains the basic encoder-decoder architecture and three contrastive learning loss functions. The architecture of our model is shown in Figure 2.

3.1 Basic Architecture for Disentanglement

Like previous disentanglement methods (Higgins et al., 2017; John et al., 2019; Sha and Lukasiewicz, 2021), we use an autoencoder as our basic architecture. Autoencoders are able to map the input

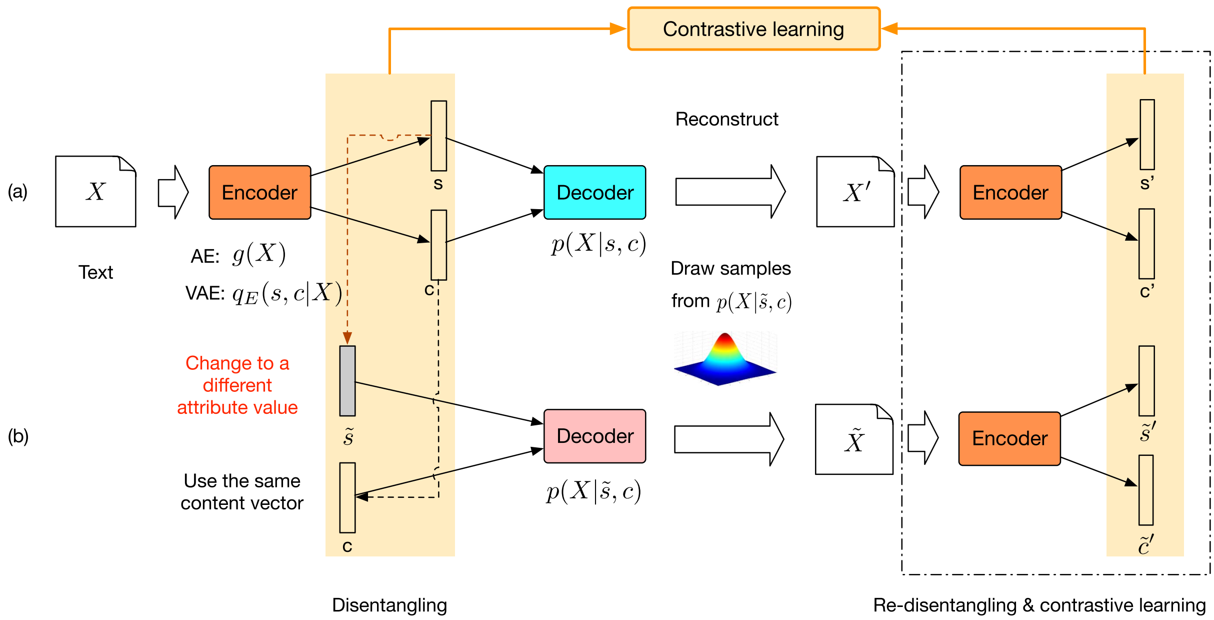


Figure 2: Complete architecture of our proposed model CLD. The upper row (a) represents the normal disentanglement process. The lower row (b) imitates the style/attribute transfer process. In both processes, we conduct re-disentanglement and use contrastive learning to encourage the content vector (c) to stay unchanged, while the style vectors (s, $\tilde{s}$) change to the desired values.

text into a latent space, while encouraging the latent vector to contain the complete information of the input. So, disentanglement is usually achieved by adding constraints to the latent space to split it into irrelevant segments. Then, each segment represents an isolated feature of the input, and once changed the reconstructed text should also be changed correspondingly.

For explicit disentanglement (with annotated attributes for training), we use two kinds of autoencoders: vanilla autoencoders (Hinton and Zemel, 1994) and variational autoencoders (VAEs) (Kingma and Welling, 2014). Given a text dataset $S_X = \{X_1, \ldots, X_N\}$, the loss functions of these two autoencoders are defined as follows:

$$J_{\mathrm{AE}} = -\,\mathbb{E}_{X \sim S_X} p(X|f(X)), \tag{1}$$

$$\begin{aligned} J_{\mathrm{VAE}} = &-\,\mathbb{E}_{z \sim q_E(z|X)} \log[p(X|z)] \\ &+ \lambda_{\mathrm{KL}} \mathrm{KL}(q_E(z|X)||p(z)), \end{aligned} \tag{2}$$

where $f(\cdot)$ and $q_E(z|X)$ are the encoders in the vanilla and the variational autoencoders, respectively, $p(X|z)$ is the decoder, and $p(z)$ is a prior distribution (usually, $\mathcal{N}(0, 1)$). The detailed architecture is given in the appendix.

Note that J_{VAE} has the name ‘‘VAE’’ because the latent space is calculated using the same method as a variational autoencoder (VAE). Specifically, a VAE uses an encoder to generate a distribution over the latent space, and then samples a vector z from this distribution, and then feeds z to a decoder. Sampling from a distribution results in a continuous latent space (Bowman et al., 2016).

3.2 Contrastive Learning for Explicit Disentanglement

Contrastive learning is originally proposed to learn such an embedding space in which similar sample pairs stay close to each other, while dissimilar ones are far apart. So, for disentangled representations, we can re-disentangle the reconstructed input and conduct contrastive learning between the disentangled representations and re-disentangled representations. Intuitively, after one disentangled feature is changed, the corresponding re-disentangled feature should also be changed, and the other re-disentangled features should remain unchanged.

Basics for Explicit Disentanglement. In explicit disentanglement, the most typical way is to separate the latent space into two irrelevant latent spaces, one for the style (s) and one for the content

(c) (John et al., 2019; Sha and Lukasiewicz, 2021). The style[1] vector here is the representation of one of the attributes of the text, including sentiment, tense, and tone for text. In this paper, we define a new symbol to represent the disentanglement: ''$\twoheadrightarrow$''. Then, $X \twoheadrightarrow [s, c]$ represents that the representations of s and c are obtained by directly splitting the latent vector z (in Eq. (2)) into s and c. On the other hand, we define ''$\rightarrowtail$'' for generating text according to the disentangled attributes. By Eq. (2), the distribution of the generated text is calculated by $p(X|s, c)$. So, we can take a sample text from this distribution as $X' \sim p(X|s, c)$, which is denoted by $[s, c] \rightarrowtail X'$ in this paper. Then, the disentanglement process and the reconstruction process are written as:

$$X \twoheadrightarrow [s, c], \qquad [s, c] \rightarrowtail X', \tag{3}$$

where X' represents the reconstructed text.

Re-disentanglement for Style Transfer. Following the unified distribution-control (UDC) method in Sha and Lukasiewicz (2021), we also predefine a Gaussian distribution $\mathcal{N}_i$ for the i-th style type value. To give a specific example, there are two values for text sentiment (positive and negative), each corresponds to a Gaussian distribution.

To directly model the style transfer process, we first change the style vector s to the vector of a different style, which is sampled from the unified style distribution defined by the UDC method. In the training phase, this sampling process can be conducted by the reparameterization trick as shown in Kingma and Welling (2014). Then, we reconstruct the text and disentangle the text for a second time (namely, re-disentangle) into style vector and content vector.

In detail, assuming that there are V possible style values for s, we sample v style values $\tilde{s}_1, \ldots, \tilde{s}_v$ that are different from s's original style value. Then, we replace s with $\tilde{s}$[2] and generate the text $\tilde{X}$. After that, we re-disentangle the generated text X' (in Eq. (3)) and $\tilde{X}$, and compare the re-disentangled representation of style and content with the original representation of style and content.

So, the generation and re-disentanglement process can be described as follows:

$$[s, c] \rightarrowtail X', \quad X' \twoheadrightarrow [s', c']; \tag{4}$$
$$[\tilde{s}, c] \rightarrowtail \tilde{X}, \quad \tilde{X} \twoheadrightarrow [\tilde{s}', \tilde{c}']. \tag{5}$$

Contrastive Learning. First, under the UDC setting, assume that the predefined trainable distributions for each style value are $\mathcal{N}_1, \ldots, \mathcal{N}_V$. The disentangled style vector s is expected to be close to the corresponding style value's representation s_*^{pre}[3] and far away from other style values' representation. Consistent with previous work (He et al., 2020), we use the dot product to measure the similarity and the InfoNCE (Oord et al., 2018) loss as the contrastive learning loss function as follows:

$$s_i^{\text{pre}} \sim \mathcal{N}_i, \quad i \in \{1, \ldots, v\}, \tag{6}$$
$$L_{\text{ori}} = -\log \frac{\exp(s \cdot s_*^{\text{pre}}/\tau)}{\sum_{i=0}^{v} \exp(s \cdot s_i^{\text{pre}}/\tau)}, \tag{7}$$

where τ is a temperature hyperparameter (He et al., 2020).

When we re-disentangle the reconstructed text as $X' \rightarrowtail [s', c']$, the representation for style s' should be close to the original style value, and far away from all the other style value's representations. The corresponding InfoNCE (Oord et al., 2018) loss is as follows:

$$L_{\text{re}} = -\log \frac{\exp(s' \cdot s_*^{\text{pre}}/\tau)}{\sum_{i=0}^{v} \exp(s' \cdot s_i^{\text{pre}}/\tau)}. \tag{8}$$

On the other hand, when the style transfer process is conducted as Eq. (5), ideally, the re-disentangled style representation $\tilde{s}'$ should be far from the original style s and close to the transferred style vector $\tilde{s}$. So, the InfoNCE (Oord et al.,

[1]Following the glossary by Sha and Lukasiewicz (2021), a **style type** is a style class that represents a specific feature of text or an image, e.g., sentiment, tense, or face direction; and a **style value** is one of the different values within a style type, e.g., sentiment (positive/negative), or tense (past/now/future).

[2]The subscript is omitted, since we do the same operation for each style type value sample.

[3]s_i^{pre} is sampled from the distribution $\mathcal{N}_i$. s_*^{pre} is sampled from the distribution $\mathcal{N}_*$, which corresponds to the ground truth attribute label.

2018) loss function for each of the sampled style values, namely, $\tilde{s}_k$ ($k = 1, \ldots, v$), is as follows:

$$\tilde{L}_k = -\log \frac{\exp(\tilde{s}'_k \cdot \tilde{s}_k/\tau)}{\exp(\tilde{s}'_k \cdot s/\tau) + \sum_{i=0, i\neq k}^{v} \exp(\tilde{s}'_k \cdot \tilde{s}_i/\tau)}, \tag{9}$$

For the re-disentangled content representations c' and $\tilde{c}'$, it should be close to the original content representation c and far from the content representation disentangled from other examples. The InfoNCE loss for content representation is $L_c(c')$. Similarly, the contrastive learning constraint for $\tilde{c}'$ is $L_c(\tilde{c}')$ as follows.

$$L_c(c') = -\log \frac{\exp(c' \cdot c/\tau)}{\sum_{i=0}^{M} \exp(c' \cdot c^{(i)}/\tau)}, \tag{10}$$

$$L_c(\tilde{c}') = -\log \frac{\exp(\tilde{c}' \cdot c/\tau)}{\sum_{i=0}^{M} \exp(\tilde{c}' \cdot c^{(i)}/\tau)}, \tag{11}$$

where $c^{(i)}$ is the disentangled content representation of the i-th example in the current batch, M represents the batch size.

Finally, if we are using a vanilla autoencoder as the basic architecture, the total loss function of contrastive-learning-based explicit disentanglement is shown in Eq. (12).

$$L = J_{\text{AE}} + \lambda_{\text{ori}} L_{\text{ori}} + \lambda_{\text{re}} L_{\text{re}} + \lambda_k \sum_{k=1}^{v} \tilde{L}_k + \lambda_c L_c, \tag{12}$$

where λ_{ori}, λ_{re}, λ_k, and λ_c are hyperparameters. When we are using a VAE as the basic architecture, we only need to replace J_{AE} with J_{VAE} in Eq. (12). L_c is obtained by summing up the two contrastive learning losses for content preservation as shown in Eq. (13). The coefficients of the three items are set to the same, because they are expected to provide an equal effect on the three latent spaces: the original latent space, the re-disentangled latent space, and the style-transferred re-disentangled latent space.

$$L_c = L_c(c') + L_c(\tilde{c}'). \tag{13}$$

4 Experiments

4.1 Data

Consistent with previous work, we use Yelp Service Reviews[4] (Shen et al., 2017), Amazon Product Reviews[5] (Fu et al., 2018), and the GoEmotions dataset[6] (Demszky et al., 2020) as the datasets for explicit disentanglement. In the Yelp dataset, there are $444k$, $63k$, and $126k$ reviews in the train, valid, and test sets, while the Amazon dataset contains $559k$, $2k$, and $2k$, respectively. Both datasets contain sentiment labels with two possible values (''pos'' and ''neg''). Additionally, the tense label is also available in the Amazon dataset, which contains three possible values (''past'', ''now'', and ''future'').

GoEmotions dataset contains 58,009 examples with the train, test, and validation sets split as 43,410, 5,427, and 5,426 examples, respectively. GoEmotions annotations categorize the examples into 27 distinct emotion labels. These emotion labels are further grouped in two ways: First, by sentiment into positive, negative, and ambiguous classes. Second, by Ekman's emotion taxonomy which divides the emotions into 6 broad categories: anger (including anger, annoyance, disapproval), disgust, fear (including fear and nervousness), joy (covering all positive emotions), sadness (including sadness, disappointment, embarrassment, grief, and remorse), and surprise (spanning all ambiguous emotions). The mapping relations are shown in Table 1.

4.2 Evaluation Metrics

We borrow the metric mutual information gap (MIG) in Chen et al. (2018) for evaluating the disentanglement performance. MIG was originally proposed for implicit disentanglement, which takes each single dimension (a scalar latent variable) of the latent vector as an attribute. In the original design, MIG measures the difference of two mutual information values, one of them is the mutual information between the ground truth factor v_k and latent variable z_* (z_* is the best fit latent variable for v_k with the largest mutual information), the other is the mutual information between the ground truth factor v_k and latent variable z_{**} (z_{**} is the second best fit latent variable for v_k). MIG is defined as follows (Chen et al., 2018):

$$\text{MIG}_{\text{im}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{H(v_k)} \Big(I(z_*; v_k) - I(z_{**}; v_k) \Big), \tag{14}$$

[4] https://github.com/shentianxiao/language-style-transfer.

[5] https://github.com/fuzhenxin/textstyletransferdata.

[6] https://github.com/google-research/google-research/tree/master/goemotions.

Sentiment:	positive	negative				ambiguous
Ekman:	joy	fear	sadness	disgust	anger	surprise
Emotions:	joy, amusement, approval excitement, gratitude, love, optimism, relief, pride, admiration, desire, caring	fear, nervousness	disappointment, embarrassment, sadness, grief, remorse	disgust	anger, annoyance, disapproval	surprise, realization, confusion, curiosity

Table 1: Mapping of emotion categories to sentiment and Ekman taxonomy in GoEmotions dataset.

where the subscript ''im'' stands for implicit disentanglement, and the mutual information $I(z; v_k)$ is defined by:

$$I(z; v_k) = \mathbb{E}_{q(z,v_k)}\Big[\log \sum_{X \in \chi_{v_k}} q(z|X)p(X|v_k)\Big], \tag{15}$$

where K is the latent vector's dimension, $H(v_k)$ is the entropy of v_k, and χ_{v_k} is the support of $p(X|v_k)$.

When computing MIG in explicit disentanglement, we replace the latent variables z_* and z_{**} by s and c:

$$\text{MIG}_{\text{ex}} = \frac{1}{H(v_k)}\Big(I(s; v_k) - I(c; v_k)\Big), \tag{16}$$

where the subscript ''ex'' stands for explicit disentanglement.

When evaluating the attribute control performance, we have 4 metrics for the NLP tasks.

- Attribute transfer accuracy (TA): Following previous works (John et al., 2019; Sha and Lukasiewicz, 2021), we use an external sentence classifier (TextCNN [Kim, 2014]) to measure the sentiment accuracy after the attribute change. The external sentence classifiers are trained separately for the Yelp and the Amazon dataset, and achieved an acceptable accuracy on the validation set (Yelp: 97.68%, Amazon: 82.32%).

- Content preservation BLEU (CBLEU-1 & CBLEU-4): This metric is proposed in Logeswaran et al. (2018), which transfers the attribute-transferred sentence back to the original attribute, and then computes the BLEU score with the original sentence.

- Perplexity (PPL): Perplexity is used for evaluating the fluency of the generated sentences. We use a third-party language model (Kneser and Ney, 1995, KenLM) as the evaluator. Two separate KenLMs are trained and used for evaluation on the two datasets.

- Transfer BLEU (TBLEU): The BLEU score is calculated between the original sentence and the attribute-transferred sentence. We delete the sentiment words before evaluation according to a sentiment word list.[7]

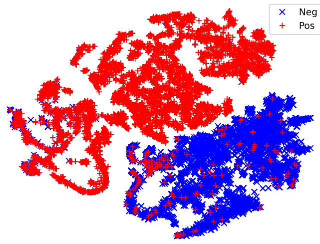
- Geometric mean (GM): We use the geometric mean of TA, 1/PPL, and TBLEU as an aggregated score, which considers attribute control performance and fluency simultaneously.

4.3 Disentanglement Performance

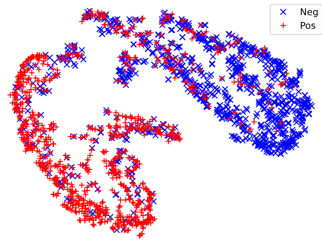
We have visualized the latent space of attributes and contents in Figures 3 and 4. To generate this visualization, we perform dimension reduction on the hidden attribute representations in the latent space. Specifically, we use t-SNE (van der Maaten and Hinton, 2008) to reduce the high-dimensional attribute representations to 2D embeddings that can be plotted. We see that with contrastive learning, both the vanilla and the variational autoencoder have separated different labels of sentiment (or tense) into different latent spaces successfully. In comparison, the different labels are mixed together in the content's latent space according to Figure 4, which means that the content space does not contain information of the sentiment attribute. Note that we do not use any resource-consuming traditional disentanglement methods like adversarial methods or mutual information minimization, simply re-disentangling the generated sentence and using contrastive learning can lead to such a good disentanglement performance.

For datasets with more granular emotion categories, we also visualize the attribute latent space of the GoEmotions dataset. We again use t-SNE

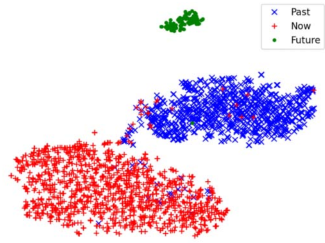
[7] `https://ptrckprry.com/course/ssd/data/positive|negative-words.txt.`



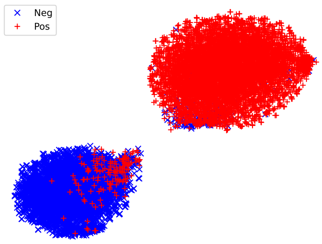
(a) Yelp (Vanilla)



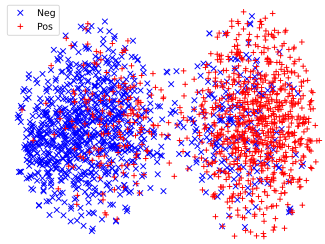
(b) Amazon (Sentiment, Vanilla)



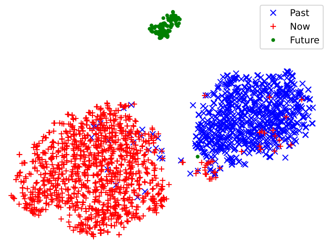
(c) Amazon (Tense, Vanilla)



(d) Yelp (VAE)

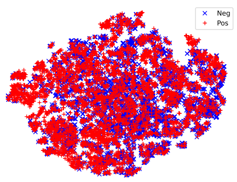


(e) Amazon (Sentiment, VAE)

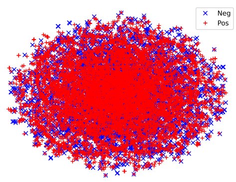


(f) Amazon (Tense, VAE)

Figure 3: Visualization of the disentangled latent space for the two style types: sentiment and tense. (a), (b), and (c) are created by a vanilla autoencoder, while (d), (e), and (f) are created by a VAE. All results are obtained when τ is set to 100.

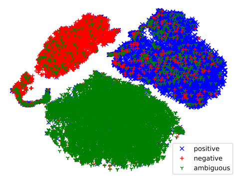


(a) Vanilla

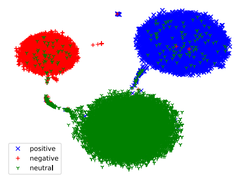


(b) VAE

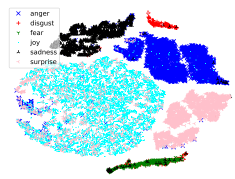
Figure 4: Visualization of the content spaces after disentanglement on the Yelp dataset.



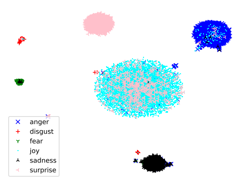
(a) Sentiment (Vanilla)



(b) Sentiment (VAE)



(c) Ekman (Vanilla)

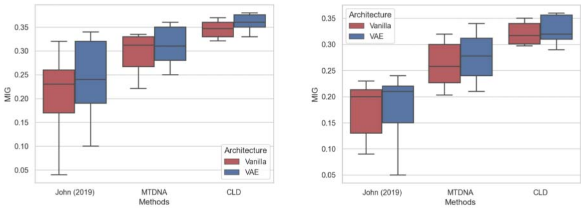


(d) Ekman (VAE)

Figure 5: Visualization of the Sentiment and Ekman taxonomy in the latent space on the GoEmotions dataset.

to reduce the high-dimensional attribute representations to 2D embeddings that can be plotted. As shown in Figure 5, the 2D latent space naturally separates into three distinct clusters corresponding to the semantic-level taxonomy of positive, negative, and neutral emotions. Furthermore, within the positive and negative regions, the space separates into smaller sub-clusters representing each of the six Ekman emotions. This demonstrates that our model has learned a disentangled latent space where proximity aligns with annotated emotion similarities. By visualizing the latent space in 2D, we can better understand the relationships learned between different emotion categories.

Also, the comparison of the MIG value is shown in Figure 6. We reimplemented the previous works of explicit disentanglement (John et al., 2019) and MTDNA (Sha and Lukasiewicz, 2021), based on their released code, the hyperparameters of the encoder and the decoder are all set to the same.



(a) Yelp dataset (b) Amazon dataset

Figure 6: The box plot of the MIG metric for different explicit disentanglement methods on the two datasets in the experiments (for sentiment). The red boxes represent vanilla-autoencoder-based methods, while the blue boxes are for VAE-based methods.

Different experiments for a model would have multiple different MIG values due to different random initialization. So, we draw box plots to show the statistical comparison of MIG values in 40 experiments. In both datasets for explicit disentanglement, our method CLD achieves a better MIG value and is more robust (has smaller variance) than the other two methods.

Additionally, due to the computational efficiency of contrastive learning losses, our proposed method takes less time for each epoch compared to adversarial-based and mutual-information-based methods. On Yelp, it takes CLD 20.93 min (Vanilla) and 21.56 min (VAE) for one epoch, while John et al. (2019) requires 46.36 min (Vanilla) and 44.59 min (VAE) for one epoch, and MTDNA (Sha and Lukasiewicz, 2021) requires 42.74 min (Vanilla) and 43.62 min (VAE) for one epoch.

4.4 Performance of Attribute Control

We compare our method CLD with multiple previous attribute control methods: Logeswaran et al. (2018) and Lample et al. (2019) as non-disentanglement methods, and John et al. (2019) and MTDNA (Sha and Lukasiewicz, 2021) as explicit disentanglement methods. We also compared our approach with the prefix-tuning-based method by Qian et al. (2022) for controlling the attribute of generated text. However, we note that their method was not specifically designed to maintain the text content while modifying attributes. Therefore, we limited our comparison to the TA and PPL metrics.

The overall performances of the Yelp and Amazon datasets are listed in Table 2. The overall performance of GoEmotions dataset is listed in Table 3. We can see that our proposed method CLD outperforms all the previous works in the TA metric, perplexity, and TBLEU score. Compared with the baseline methods without contrastive learning, our approach shows great advantages over the MTDNA (Sha and Lukasiewicz, 2021) models in the CBLEU metrics. This fact shows that the content of a sentence is much easier to be preserved (the attribute control process is more robust) when we are using contrastive learning to keep the content vector before and after re-disentanglement to be as close as possible. Moreover, when we added back-translation loss as is conducted by Logeswaran et al. (2018) and Lample et al. (2019), our method CLD achieved an even higher score in the CBLEU-1 and CBLEU-4 metric, and this score has outperformed the state-of-the-art CBLEU score. This again proved that back-translation loss will become more powerful in content preservation when used together with contrastive learning. According to the aggregated performance (GM) listed in Table 2, CLD also outperforms the baseline methods, and CLD-(VAE) with back-translation loss achieved state-of-the-art results. We have observed similar results in the tense attribute, which is shown in the column ‘‘TA(T)’’ in Table 2.

We also conducted a comparison between our method and the prompt-tuning-based approach proposed by Qian et al. (2022). However, it is important to note that the prompt-tuning-based method only focuses on controlling the attribute of the generated text, without ensuring content preservation. Therefore, we limited our comparison to the TA and PPL metrics. To evaluate their work, we applied Qian et al.’s (2022) method on our datasets and assessed the results based on our metrics. As demonstrated in Table 2, our method still has a clear advantage over the prompt-tuning-based approach, as the latter sacrifices some attribute accuracy in order to achieve controllable text generation.

Our method is very easy to be merged with pretrained language models in encoder-decoder architectures (like T5 [Raffel et al., 2020]). We merged our method with T5 and report the results in Table 2. Due to the large storage of text corpus and common sense knowledge in the pretrained language model, the result achieved a much better level in style transfer accuracy, content preservation, and fluency metrics.

	Yelp						Amazon						
	TA	CBLEU-1	CBLEU-4	PPL	TBLEU	GM	TA(S)	TA(T)	CBLEU-1	CBLEU-4	PPL	TBLEU	GM
(Logeswaran et al., 2018)	0.905	53.0	7.5	133	17.4	0.105	0.857	–	31.5	1.8	187	16.6	0.091
(Lample et al., 2019)	0.877	–	–	48	14.6	0.139	0.896	–	–	–	92	18.7	0.122
(John et al., 2019) (Vanilla)	0.883	–	–	52	18.7	0.147	0.720	–	–	–	73	16.5	0.118
(John et al., 2019) (VAE)	0.934	–	–	32	17.9	0.174	0.822	–	–	–	63	9.8	0.109
MTDNA (Vanilla)	0.877	30.4	4.3	45	16.1	0.146	0.789	0.963	23.4	1.2	68	15.4	0.121
MTDNA (VAE)	0.944	32.6	5.1	27	21.2	0.195	0.902	0.993	24.0	1.2	44	20.1	0.160
(Qian et al., 2022)	0.873	–	–	37	–	–	0.795	0.902	–	–	65	–	–
CLD (Vanilla)	0.928	45.5	6.9	43	16.3	0.152	0.843	0.972	27.6	1.5	68	15.9	0.125
+ Back-Translation loss	0.890	54.1	**8.7**	38	16.8	0.158	0.844	0.975	36.7	2.2	59	17.1	0.135
+ T5	0.930	56.6	10.4	33	20.7	0.180	0.889	0.982	37.4	2.4	55	19.3	0.146
CLD (VAE)	**0.951**	45.7	6.3	28	22.5	0.197	**0.910**	**0.994**	28.2	1.6	43	21.3	0.165
+ Back-Translation loss	0.936	**54.3**	8.4	**26**	**22.7**	**0.201**	0.908	0.993	**37.2**	**2.3**	**40**	**21.7**	**0.170**
+ T5	0.985	58.1	11.2	25	23.7	0.211	0.921	0.994	38.3	2.5	38	22.9	0.177

Table 2: Overall attribute control performance. For the sentiment type, the transfer direction is ‘‘Neg→Pos’’, and ‘‘Pos→Neg’’. For the tense type, the transfer direction is ‘‘Past→Now’’, ‘‘Now→Future’’ and ‘‘Future→Past’’. TA(S) is the TA metric for sentiment, while TA(T) is for tense. All the advantages of our results compared to the previous best results are statistically significant, as confirmed by the Wilcoxon signed-rank test ($p < 0.05$). The state-of-the-art results made by pretrained language models are underlined.

	TA(Sentiment)	TA(Ekman)	CBLEU-1	CBLEU-4	PPL	TBLEU	GM-4
(Logeswaran et al., 2018)	0.723	0.538	21.2	1.5	224	8.9	0.111
MTDNA (Vanilla)	0.759	0.602	25.4	3.1	136	9.5	0.134
MTDNA (VAE)	0.780	0.635	28.6	3.7	95	12.1	0.158
(Qian et al., 2022)	0.852	0.816	–	–	97	–	–
CLD (Vanilla)	0.864	0.845	34.9	4.6	79	15.4	0.194
+ Back-Translation loss	0.857	0.832	36.5	5.2	**71**	17.8	0.206
+ T5	0.893	0.887	39.7	7.1	63	20.3	0.225
CLD (VAE)	**0.899**	**0.896**	36.1	5.5	76	19.4	0.211
+ Back-Translation loss	0.886	0.858	**37.3**	**6.6**	74	**21.5**	**0.217**
+ T5	0.923	0.901	39.6	8.2	60	23.3	0.238

Table 3: Overall attribute control performance of GoEmotions dataset. For the sentiment taxonomy, the transfer direction is ‘‘Negative→Positive’’, and ‘‘Positive→ambiguous’’. For the Ekman taxonomy, the transfer direction is ‘‘joy→fear’’, ‘‘fear→sadness’’, ‘‘sadness→disgust’’, ‘‘disgust→anger’’, ‘‘anger→surprise’’, ‘‘surprise→joy’’. TA(Sentiment) is the TA metric for sentiment, while TA(Ekman) is for Ekman taxonomy. All the advantages of our results compared to the previous best results are statistically significant, as confirmed by the Wilcoxon signed-rank test. ($p < 0.05$). The state-of-the-art results made by pretrained language models are underlined.

4.5 Ablation Test

Effect of the Re-disentanglement Process. To prove that the re-disentanglement process is necessary, we remove all the contrastive losses related to the re-disentanglement process. The visualization of the latent spaces for vanilla and VAE are shown in Figure 8. We can see that the latent space became partly mixed up, which shows that the re-disentanglement process is indispensable.

Effect of Contrastive Loss Functions. To study the effect of each contrastive learning loss, we remove the loss functions one by one to check the difference of the evaluation metrics. The results are shown in Table 4. We found that after the

	Yelp		
	TA	CBLEU-4	TBLEU
CLD (Vanilla)	0.928	**6.9**	16.3
CLD (Vanilla) - L_c	0.935	4.6	11.5
CLD (Vanilla) - L_c -$\tilde{L}_k$	0.903	4.3	10.8
CLD (Vanilla) - L_c -$\tilde{L}_k$-L_{re}	0.862	4.4	10.2
CLD (VAE)	0.951	6.3	**22.5**
CLD (VAE) - L_c	**0.959**	4.2	13.6
CLD (VAE) - L_c -$\tilde{L}_k$	0.928	4.3	12.8
CLD (VAE) - L_c -$\tilde{L}_k$-L_{re}	0.887	4.1	12.4
CLD (Vanilla) (MSE)	0.926	5.0	12.2
CLD (VAE) (MSE)	0.945	5.1	15.6

Table 4: Ablation test results. We select three metrics (TA, CBLEU-4, and TBLEU) in this experiment, because they are closely related to the contrastive losses L_{re}, $\tilde{L}_k$, and L_c.

content contrastive loss L_c is removed, the style transfer accuracy is improved, which shows that the constraint on the content vector would negatively affect the style information in the generated sentences. Also, the CBLEU-4 and TBLEU scores largely dropped, which shows that L_c is very important for content preservation. Then, after $\tilde{L}_k$ is removed, the TA metric dropped about 3 percentage points, while the CBLEU-4 and TBLEU scores did not have any significant change. Since $\tilde{L}_k$ is a constraint for the re-disentangled style vector of the style-transferred sentence, it does not have too much effect on the content of the sentence. A similar phenomenon is observed when we remove the loss L_{re}: The TA metric significantly decreased again, and the BLEU scores slightly decreased.

Additionally, we also remove the three contrastive learning losses for the content preservation ($L_c(c')$, $L_c(\tilde{c}')$) to study their effect on the results. The scores are also listed in Table 5. We can see that removing any one of the two losses would cause an increase in the TA score, which means all of the content preservation losses are limitations on the style latent space. Both the CBLEU-4 and TBLEU scores decrease a lot after removing the two content preservation losses. In particular, it seems that $L_c(c')$ has the largest effect on the scores, which is sensible, because a more distinguishable content space is easier for content preservation intuitively.

	Yelp		
	TA	CBLEU-4	TBLEU
CLD (Vanilla) - $L_c(c')$	0.929	5.2	14.8
CLD (Vanilla) - $L_c(\tilde{c}')$	0.930	6.1	15.3
CLD (VAE) - $L_c(c')$	0.955	5.1	17.8
CLD (VAE) - $L_c(\tilde{c}')$	0.951	5.8	20.1

Table 5: Ablation test results w.r.t. different components in L_c.

We also conducted experiments about changing the content's contrastive learning loss L_c to mean-square error (MSE) loss to check whether contrastive learning is necessary. In this experiment, we replace L_c with the following loss L_{mse}:

$$L_{\text{mse}} = \|c' - c\|_2 \,, \tag{17}$$

where $\|\cdot\|_2$ represents the 2-norm. The results are also shown in line CLD (Vanilla) (MSE) and CLD (VAE) (MSE) of Table 4. We can see that the score of CBLEU-4 and TBLEU dropped considerably compared with CLD (Vanilla) and CLD (VAE) after we replaced L_c with L_{mse}. The intrinsic difference between L_c and L_{mse} is that L_{mse} only encourages c' and c from the same case to be close, while L_c also requires the content vectors from different cases to be far away form each other. The latter alleviates the possibility of the content space to collapse. This result proved that the contrastive learning loss is inevitable for content preservation.

Effect of τ. To investigate the effect of the temperature hyperparameter τ, we run the model several times with different values of τ, and visualize the latent space in Figure 7. According to Figure 7, when τ has a small value, the latent spaces for the different style values tend to be connected in some area. In contrast, the latent spaces become separated when the value of τ increases. The reason is that when the temperature τ is getting large, the distinction between the positive and negative examples in the contrastive losses tends to be underestimated. Hence, the model needs to work harder to make the distinction large, and thus the latent spaces are getting more separated.

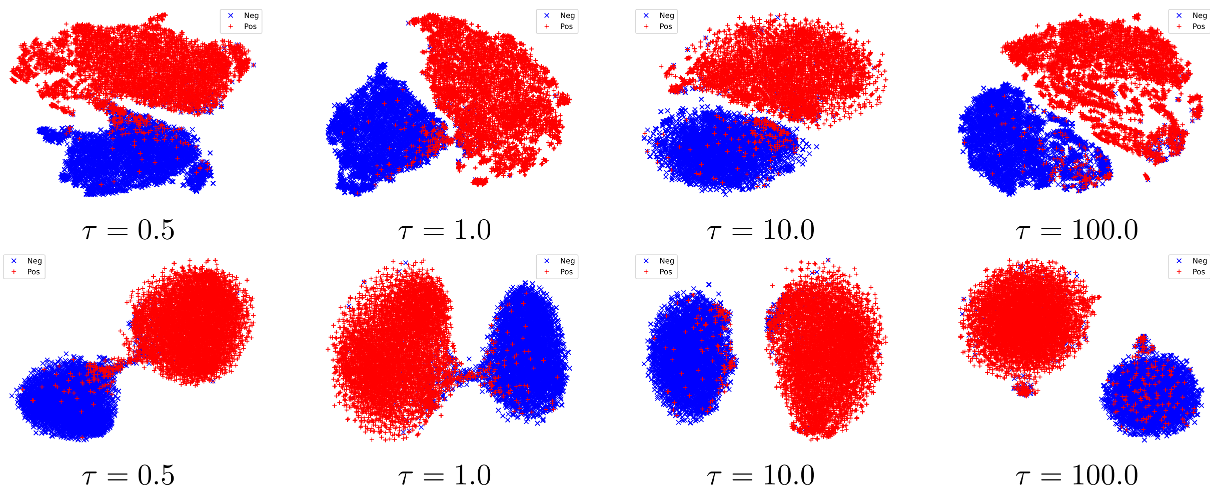
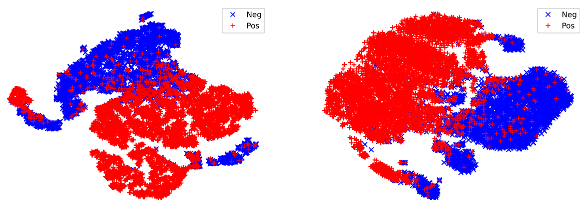


Figure 7: Change of the latent space when the temperature hyperparameter τ is getting larger. We show four different τ values (i.e., $0.5, 1.0, 10.0, 100.0$) for the two possible architectures. The first row is from the vanilla autoencoder architecture, while the second row is from the VAE architecture.



(a) Yelp (Vanilla) (b) Yelp (VAE)

Figure 8: Visualization of latent space when we remove the re-disentanglement process, i.e., we only keep the contrastive losses in Eqs. 9.

4.6 Case Study

We sampled some generated text when we are transferring the sentiment attribute from one to another, the results are shown in Table 6. The corresponding results for tense are shown in Table 7. According to the results, the content of text almost remains unchanged, while the target attribute was changed to what we expected.

Furthermore, we evaluated more complex emotion attribute transfer cases from the GoEmotions dataset. We transformed the emotions according to the Ekman taxonomy and presented the results produced by CLD using both the vanilla and VAE architectures. These results are tabulated in Table 8.

4.7 Human Evaluation

We also conducted a human evaluation for the attribute control results. We sampled 1,000 examples from each of Yelp and Amazon, and changed their attribute value to the opposite value (''Positive''→''Negative'', ''Negative''→''Positive''). Then, we collected the generated sentences and asked 3 data graders to give a score to the sentences on 3 metrics (transfer accuracy (TA), content preservation (CP), and language quality (LQ)). Among them, TA is a percentage, CP and LQ are scored between $1 \sim 5$. The detailed questions are listed in the appendix. We randomly shuffled the sentences to remove the ordering hint. The final result of human evaluation is shown in Table 9. The inter-rater agreements (the Krippendorff alpha values) of the three metrics are 0.84, 0.89, and 0.92, all of them are acceptable due to Krippendorff's principle (2004). We can see that our proposed method CLD outperforms the baseline in each of the human evaluation metrics. We also listed some generated cases in Appendix 4.6.

5 Discussion

Recent work has explored utilizing large language models (LLMs) like ChatGPT and GPT-4 for controllable text generation. For example, Reif et al. (2021) have proposed methods to steer text style transfer in these LLMs by conditioning on discrete attributes or continuous latent representations. Compared to our approach, a key difference is that we train our model end-to-end to disentangle latent attributes, while LLMs rely on prompting or fine-tuning approaches applied post-hoc.

Original (Pos)	Vanilla Transferred (Neg)	VAE Transferred (Neg)
every one is so nice, and the food is amazing !	the servant is rude and the food is terrible .	every one is so tepid, and the food is awful.
an excellent dining experience .	the dining feels bad .	an awful dining experience .
yesterday i went to this location and the staff was very informative and personable .	yesterday i went to this location and found the staff so rude and angry .	yesterday i went here and the staff was very tepid, not a good choice .
Original (Neg)	**Vanilla Transferred (Pos)**	**VAE Transferred (Pos)**
crap service with mediocre food is not a good business model to live by .	good service and the food is delicious .	good service with delicious food, good business model to live by .
this is a horrible representation of a deli .	this is a great place to go in this area .	this is a good place of a deli .
the staff does a horrible job with my teenagers .	the staff works well with my teenagers .	the staff does a great job working with my teenagers.

Table 6: Examples of sentiment polarity control.

Original (Now)	Vanilla Transferred (Past)	VAE Transferred (Past)
this machine is exactly what the name says it is - a speller .	this machine was exactly a speller .	The machine was a speller, just as its name indicated.
it's so small (of course) and it's really only good for nuts .	it was so small and only good for nuts .	it was very small and only useful for nuts in the past, just as it is now .
Original (Past)	**Vanilla Transferred (Future)**	**VAE Transferred (Future)**
i did not like the taste of this at all.	i will never like this taste .	i will never like this taste any more .
i was not impressed, but at least i tried.	I will never be impressed .	I will not be impressed, but at least I will try.
Original (Future)	**Vanilla Transferred (Past)**	**VAE Transferred (Past)**
i'm going to e-mail the company but in the meantime, if you drink this tea, stop.	I emailed the company .	I emailed the company, stop drinking this tea .
i'm probably going to end up throwing all of these out .	I threw all this out probably .	I probably ended up throwing all of these out.

Table 7: Examples of tense control.

While promising, utilizing LLMs for attribute-controlled generation remains challenging. The discrete prompting approach can yield brittle or superficial style changes, as the models' understanding of prompted attributes is imperfect and limited to correlation patterns in the pretraining data (Reif et al., 2021; Luo et al., 2023). Latent space steering has shown more coherent style transfer, but current methods rely on complex optimization schemes or assume access to an attribute classifier (John et al., 2019; Sha and Lukasiewicz, 2021). In contrast, our model learns disentangled representations directly from data through closed-loop contrastive training.

6 Limitations

Controlling the Attribute's Intensity Our model is not designed to control the intensity of an attribute, like generating some neutral sentence instead of ‘‘pos’’ or ‘‘neg’’. If we want to generate a neutral sentence anyway, we just need to take the average vector of the mean value of the ‘‘pos’’ and ‘‘neg’’, and replace the original semantic style vector. Then, the decoder will generate a neutral sentence. However, this method will not always be successful, because there is no guarantee that these latent spaces are smoothly distributed with overlapping regions, and the

Original	**Vanilla Transferred**	**VAE Transferred**
My friend and I have been having good fun queuing Mei/Winston, trololo, and 2 bottles of whisky. (Joy)	My friend and I have been anxiously queueing Mei/Winston, trololo, and 2 bottles of whisky (fear)	My friend and I anxiously wait in the endless, crawling queue for Mei/Winston, trololo, and 2 bottles of whisky. (fear)
And then you're stuck on that medication for life, sounds like a scary trade-off. (fear)	And then I guess you're stuck taking that medicine forever and ever, seems like a real bummer of a trade-off.(sadness)	And then you're frustratedly stuck on that medication for life, sounds like a grim trade-off. (sadness)
it's hurting my heart :(why did you leave me (sadness)	It's sickening my heart :(why did you abandon me. (disgust)	It's breaking my heart :(why did you betray and leave me in such a repulsive manner. (disgust)
Gross. I hate using those reusable cloth ones because they retain smells. You've got cats? I know, I can smell them. (disgust)	Damn it. It infuriates me to use those reusable fabric ones since they hold onto scents. Throw out these cats! (anger)	Damn it. I detest using those reusable cloth ones because they retain odors. You've got cats? I know, I can detect their stench. (anger)
I do remember this and wanted nothing more than to kill that guy (anger)	Yeah I totally remember this and was like, woah, my mind was blown about that guy. (surprised)	I do recall this vividly and was utterly astonished at that guy. (surprised)
Can't believe [NAME] has been in the league that long. . . (surprised)	It's amazing that [NAME] has been in the league that long! (joy)	It's so thrilling that [NAME] has already been playing in the league for that many years! (joy)

Table 8: Examples of Ekman control in GoEmotions dataset.

		TA	CP	LQ
Yelp	(Logeswaran et al., 2018)	86.01	3.81	3.89
	(Lample et al., 2019)	82.32	3.59	4.28
	(John et al., 2019)(VAE)	85.89	3.65	4.25
	MTDNA (Vanilla)	84.28	3.69	4.32
	MTDNA (VAE)	86.04	3.78	4.39
	(Qian et al., 2022)	83.43	3.65	4.41
	CLD (Vanilla)	85.42	3.70	4.32
	CLD (VAE)	**87.98**	**3.90**	**4.43**
Amazon	(Logeswaran et al., 2018)	80.21	3.68	3.73
	(Lample et al., 2019)	77.76	3.14	3.66
	(John et al., 2019)(VAE)	82.23	3.27	3.75
	MTDNA (Vanilla)	79.03	3.34	3.74
	MTDNA (VAE)	83.28	3.52	4.08
	(Qian et al., 2022)	80.75	3.21	4.10
	CLD (Vanilla)	80.56	3.68	3.76
	CLD (VAE)	**83.96**	**3.75**	**4.32**

Table 9: Human evaluation results on Yelp and Amazon.

decoder may not have been required to generate such texts with novel style features during training. To better control the attribute's intensity, it is required to design some special mechanics in a supervised manner.

Difficult Attributes Apart from the simple text attributes, there are also some complex attributes like some specific author's style of writing, which are usually intertwined together in the latent space. Discrete categorical style types are hard to design for such kind of complex attributes. Whether disentanglement can be used for controlling complex attributes requires further research.

7 Conclusion

In this paper, we proposed a novel explicit disentanglement method, called contrastive learning disentanglement (CLD), which uses contrastive learning as the core method. Differently from previous works, we re-disentangle the reconstructed sentences, and conduct contrastive learning between the disentangled vectors and the re-disentangled vectors. To encourage the disentanglement of the attributes' latent space, we propose the re-disentangled contrastive loss L_{re} and the transferred re-disentangled contrastive loss $\tilde{L}_k$. The latter fully imitates the attribute control process. To encourage content preservation, we proposed the content contrastive loss L_c, which contains three sub-losses. These sub-losses make the content space more distinguishable and encourage the content keep unchanged during attribute control. Our proposed method is not only much easier in the mathematical derivations, it also outperforms all the compared methods in the evaluation metrics according to our experimental results.

Acknowledgments

This work was supported by the ESRC grant ES/S010424/1 ‘‘Unlocking the Potential of AI for English Law’’, by the National Natural Science Foundation of China under grants KZ37117501, ZG216S23E8, and 62306024; by the Alan Turing Institute under the EPSRC grant EP/N510129/1; and by the AXA Research Fund. We also acknowledge the use of Oxford’s Advanced Research Computing (ARC) facility, of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1), and of GPU computing support by Scan Computers International Ltd.

References

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. 2019. Unsupervised state representation learning in Atari. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 8769–8782.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203. https://doi.org/10.18653/v1/P19-1019

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *International Conference on Learning Representations*. https://doi.org/10.18653/v1/D18-1399

Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 32:15535–15545.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. https://doi.org/10.18653/v1/K16-1002

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *34th Conference on Neural Information Processing Systems (NeurIPS)*.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3). https://doi.org/10.1007/978-3-642-02172-5_2

Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. 2020b. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020c. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*. https://doi.org/10.1109/ICCV48922.2021.00950

Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka, Kwan Ho Ryan Chan, Pengyuan Zhai, Yaodong Yu, Xiaojun Yuan, Heung-Yeung Shum, and Yi Ma. 2022. Ctrl: Closed-loop transcription to an ldr via minimaxing rate reduction. *Entropy*, 24(4). https://doi.org/10.3390/e24040456, PubMed: 35455120

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*. https://doi.org/10.18653/v1/2020.acl-main.372

Joseph J. Di Stefano, Allen R. Stubberud, and Ivan Williams. 1967. *Feedback and Control Systems*.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N. Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem van de Meent. 2018. Structured disentangled representations. *arXiv preprint arXiv:1804.02086*.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Yan Rui. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v32i1.11330

Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. 2020. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Michael U. Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2).

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE. https://doi.org/10.1109/CVPR.2006.100

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738. https://doi.org/10.1109/CVPR42600.2020.00975

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. Beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2(5):6.

Geoffrey E. Hinton and Richard S. Zemel. 1994. Autoencoders, minimum description length, and Helmholtz free energy. *Advances in Neural Information Processing Systems*, 6:3–10.

R. Devon Hjelm and Philip Bachman. 2020. Representation learning with video deep infomax. *arXiv preprint arXiv:2007.13278*.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-based Pattern Recognition*, pages 84–92. Springer. https://doi.org/10.1007/978-3-319-24261-3_7

Matthew D. Hoffman and Matthew J. Johnson. 2016. ELBO surgery: Yet another way to carve up the variational evidence lower bound. In *Proceedings of the Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1041

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1181

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. *Proceedings of the International Conference on Learning Representations*.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184. https://doi.org/10.1109/ICASSP.1995.479394

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762. https://doi.org/10.18653/v1/2020.emnlp-main.55

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 2017. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *Proceedings of the International Conference on Learning Representations*.

Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3031549

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. 2020. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Towards Fine-grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/P19-1194

Guoqing Luo, Yu Tong Han, Lili Mou, and Mauajama Firdaus. 2023. Prompt-based editing for text style transfer. *arXiv preprint arXiv:2301.11997*.

Emile Mathieu, Tom Rainforth, Siddharth Narayanaswamy, and Yee Whye Teh. 2018. Disentangling disentanglement in variational autoencoders. *arXiv preprint arXiv:1812.02833*.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. 2018. Invariant representations without adversarial training. In *Advances in Neural Information Processing Systems*, pages 9084–9093.

Siddharth Narayanaswamy, T. Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. 2017. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.229

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*. https://doi.org/10.18653/v1/2022.acl-short.94

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. 2018. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. https://doi.org/10.1109/ICRA.2018.8462891

Lei Sha and Thomas Lukasiewicz. 2021. Multi-type disentanglement without adversarial training. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v35i11.17146

Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek Abdelzaher. 2020. ControlVAE: Controllable variational autoencoder. In *Proceedings of the International Conference on Machine Learning*, pages 8655–8664. PMLR.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 6833–6844.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using T-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Xiaolong Wang and Abhinav Gupta. 2015. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802. https://doi.org/10.1109/ICCV.2015.320

Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. 2020. Delving into inter-image invariance for unsupervised visual representations. *arXiv preprint arXiv:2008.11702*.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012. https://doi.org/10.1109/ICCV.2019.00610

Appendix

A Human Evaluation Questions

A.1 Transfer Accuracy (TA)

Q: Do you think the given sentence belongs to positive sentiment or negative sentiment?

- A: Positive.
- B: Negative.

A.2 Content Preservation (CP)

Q: Do you think the generated sentence has the same content with the original sentence, although the sentiment/tense is different?

Please choose a score according to the following description. Note that the score is not necessary to be integer, you can give scores like 3.2 or 4.9 by your feeling.

- 5: Exactly. The contents are exactly the same.
- 4: Highly identical. Most of the content are identical.
- 3: Half. Half of the content is identical.
- 2: Almost Not the same.
- 1: Totally different.

A.3 Language Quality (LQ)

Q: How fluent do you think the generated text is? Give a score based on your feeling.

Please choose a score according to the following description. Note that the score is not necessary to be integer, you can give scores like 3.2 or 4.9 by your feeling.

- 5: Very fluent.
- 4: Highly fluent.
- 3: Partial fluent.
- 2: Very unfluent.
- 1: Nonsense.

B Detailed Model Structure

The encoder and decoder in this paper are in the LSTM architecture. Given an input sentence $X = \{x_1, \dots, x_n\}$, the representation of the sentence z is computed by:

$$z = LSTM(x_1, \dots, x_n)\,, \tag{18}$$

where z is the output of the last LSTM cell.

In the vanilla autoencoder, z is split into a style vector and a content vector by a feed-forward layer as follows:

$$[s, c] = \tanh(W_h z + b_h)\,, \tag{19}$$

where W_h and b_h are trainable parameters.

In the variational autoencoder, the output of the encoder is further mapped into two vectors μ and $\log(\sigma^2)$. The latent vector z is sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma)$. The style vector and content vector are split by Eq. (19).

After the transfer of the style vector (s changed to s'), the style vector and content vector are merged into a new latent vector z', which is the input to the LSTM decoder as the initial state.

C Detailed Model Settings

The encoder and decoder are set as 2-layer LSTM RNNs with input dimension of 100, and the hidden size is 150. The hyperparameters are set to $\lambda_{\text{ori}} = 1.0, \lambda_{\text{re}} = 1.0, \lambda_k = 1.5, \lambda_c = 2.0, \lambda_{\text{KL}} = 0.01$ and $\tau = 100$. We used Adam for optimization, and the learning rate is set to 0.001. The model is run on an Nvidia v100 GPU.