SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2024, Vol. 67, Iss. 4, 142105:1–142105:16
https://doi.org/10.1007/s11432-022-3726-0

Memory-enhanced text style transfer with dynamic style learning and calibration

Fuqiang LIN[1], Yiping SONG[1*], Zhiliang TIAN[2], Wangqun CHEN[1], Diwen DONG[1] & Bo LIU[1,3*]

[1]*College of Computer Science and Technology, National University of Defense Technology, Changsha* 410073*, China;*
[2]*Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong* 999077*, China;*
[3]*Strategic Assessments and Consultation Institute, Academy of Military Sciences, Beijing* 100097*, China*

Received 4 March 2022/Revised 21 June 2022/Accepted 23 September 2022/Published online 26 March 2024

Abstract Text style transfer aims to rephrase a sentence to match the desired style while retaining the original content. As a controllable text generation task, mainstream approaches use content-independent style embedding as control variables to guide stylistic generation. Nonetheless, stylistic properties are context-sensitive even under the same style. For example, "delicious" and "helpful" convey positive sentiments, although they are more likely to describe food and people, respectively. Therefore, desired style signals must vary with the content. To this end, we propose a memory-enhanced transfer method, which learns fine-grained style representation concerning content to assist transfer. Rather than employing static style embedding or latent variables, our method abstracts linguistic characteristics from training corpora and memorizes subdivided content with the corresponding style representations. The style signal is dynamically retrieved from memory using the content as a query, providing a more expressive and flexible latent style space. To address the imbalance between quantity and quality in different content, we further introduce a calibration method to augment memory construction by modeling the relationship between candidate styles. Experimental results obtained using three benchmark datasets confirm the superior performance of our model compared to competitive approaches. The evaluation metrics and case study also indicate that our model can generate diverse stylistic phrases matching context.

Keywords style transfer, memory-enhanced method, text generation, deep learning, text representation

1 Introduction

As an essential task of controllable text generation, text style transfer (TST) aims to modify the stylistic attributes[1)] (e.g., sentiment, genre, and formality) of text while maintaining the underlying content. TST has been a research problem of interest due to its broad applications, such as sentiment modification [1,2], stylistic summarization [3], and text simplification [4]. Because of the difficulty in collecting parallel corpora, related research is typically conducted in the unsupervised learning setting.

To control the style attributes of transferred texts, dominant approaches learn one individual style of embedding or static latent style variables and leverage such style signals to guide transfer. One line of methods disentangles text into separated style and content representations and applies a style-specific decoder to conduct transfers conditioned on non-stylistic content and desired style embedding. Representative studies [5–8] adopt adversarial discriminators on the latent space to achieve disentanglement. Following this trend, several methods [2, 9, 10] apply pipeline word-level processing that first obtains a content-only sentence by explicitly removing stylistic tokens and then merges it with target style signals for transfer. To enhance content preservation, another line of methods [11, 12] proposes to encode text into entangled representations without explicit disentanglement and incorporates the style embedding

* Corresponding author (email: songyiping@nudt.edu.cn, kyle.liu@nudt.edu.cn)

1) We follow most existing work to regard the style concept simply as characteristics of the given corpus distinct from a general text corpus or another given style corpus.

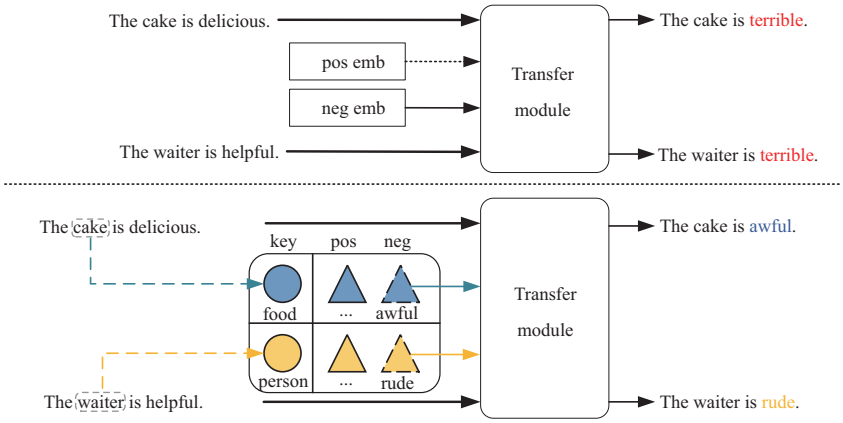

Figure 1 (Color online) Illustration of difference in style signals between most previous methods (top) and our method (bottom).

into an attention-based structure for style control. Typically, Dai et al. [11] did not assume disentanglement and apply the transformer architecture with attention mechanisms to learn style transfer, achieving considerable improvement in content preservation.

Nevertheless, as the mainstream form of style signals, simple content-independent style embedding is insufficiently expressive for projecting the concept of style, particularly being insufficiently flexible when applied to various topics of content. Because of the deep fusion of content and style in text [13], style properties, e.g., style-related phrases, vary with the content of the text [14]. Take as an example restaurant reviews with the same positive sentiment: compared with employing "great" to describe all objects, it is more reasonable to use stylistic tokens "kind", "prompt", and "delicious" to describe "owner", "service", and "food", respectively. That is, the desired style representation requires matching the semantics of the content. However, previous work overemphasizes learning content-independent style representations, which are less sensitive to the content and lead to tired style-related phrases in the transferred sentences.

To address these issues, we propose a memory-enhanced transfer method (METM) to provide fine-grained content-dependent style signals. Instead of employing individual style embedding, our method emphasizes the differences in stylistic expression under varied contents and constructs a memory to provide flexible style representations. Specifically, we divide the training corpus into multiple groups by clustering, then extract and memorize the linguistic information of each group to assist TST. As illustrated in Figure 1, the linguistic information in our memory is in the form of content-style pairs, which correspond to the common characteristics of content and style. Compared to solo-style embedding, style representations in memories are more expressive and can be retrieved dynamically with respect to content.

Moreover, because of the skew distribution of training samples, the quality of different memory units is uneven. Importantly, the memory unit with only a few samples is prone to suffer from error deviations of style representation caused by outliers. For this reason, we further introduce a calibration method to optimize the style representations in memory. Inspired by TransE [15], we consider the relationship between styles (i.e., value cells in the memory) to share a similar distribution across all memory units. Therefore, our method learns a standard relationship representation and uses it to calibrate the value cells of memory. In this way, our method can retrieve diverse stylistic phrases to match the content of input sentences, which contributes to improving content preservation and style transfer strength.

Our contributions can be summarized as follows:

(1) We extract linguistic information from the training corpus and construct a content-style memory to learn content-dependent style representations, which contributes to improving style control as well as enhancing content preservation.

(2) We propose to learn the relationship between styles and use it as shared criteria to calibrate the memory, which addresses the uneven quality of different memory units.

(3) Experimental results demonstrate that our proposed method generally outperforms state-of-the-art (SOTA) approaches on two benchmark style transfer tasks. Specifically, our model achieves a better trade-off between style transfer accuracy and content preservation and generates diverse stylistic phrases that vary with context.

2 Related work

2.1 Unsupervised TST

The method for representing content and style signals is the core topic of most existing works. Considering this criterion, we can roughly divide most previous studies into two categories: disentanglement-based methods and attention-based methods.

Disentanglement-based methods follow style transfer in the computer vision field [16] to first strip style from text and then fuse the style-independent content with the target style for transfer. Fu et al. [17] systematically explored two methods, the multi-decoder model and the style-embedding model, with the common ground to strip the style information of the input sentence using an auto-encoder model. To achieve further improvement, iterative optimization, reinforcement learning, and a series of all-around losses are applied to guarantee the quality of disentanglement [8,18,19]. In contrast to end-to-end training, attempts have also been made to adopt a two-step pipeline for word-level disentanglement. These methods identify stylistic words by pretrained discriminators and then replace them with words carrying the desired style [2,9,20,21]. Based on the assumption that stylistic properties of sentences are automatically diluted when translating to another language, another line uses unsupervised neural machine translation models with the back-translation strick [7,22]. Because of impracticable complete disentanglement, this paradigm performs well in transfer accuracy but suffers from poor content preservation [11,14]. For this reason, our work replaces the disentanglement constraint with two auxiliary tasks to make two latent representations biased toward modeling content and style information, respectively.

Instead of disentangling content and style separately, attention-based methods directly edit an entangled latent representation and rely on the generator to rewrite the original stylistic information with the desired style [11, 12, 23–25]. Specifically, Yi et al. [12] learnt latened style space from multiple instances using the generative flow method and combined it with an attention-based Seq2Seq structure to enhance content preservation. Liu et al. [25] adopted the pretrained language model GPT-2 with semantic similarity metrics as a direct reward for the stylistic generation. Generally, one important advantage of this line is an avoidance of the loss of content information caused by disentanglement, thereby better preserving content information. Nevertheless, the style-specific decoders bear all the burdens to "overwrite" the original style in the entangled latent representation, resulting in unsatisfactory style transfer strength [12, 14].

Regardless of content representation, both lines employ content-independent embeddings or static latent variables to represent style, which is insufficient because of the limited capacity of individual vectors [26]. To address this problem, Xiao et al. [26] recently proposed a transductive style transfer (TSST) model to employ a context-aware style representation by using a retriever. Our method and TSST verify that learning fine-grained content-dependent style characteristics is very helpful for altering style while retaining content. Nevertheless, there are two nontrivial differences. First, TSST only retrieves the top-k relevant samples with the target style to facilitate style representation construction, while our method takes the insights from information retrieval and integrates all training samples through clustering to learn more expressive and flexible style representations. Second, our method considers the skew distribution on semantics space and proposes to calibrate stylistic properties by modeling the relationship between styles, which potentially provides further performance gains.

2.2 Memory-enhanced text generation

Our work is also related to memory-enhanced text generation methods [20,27–29], which memorize external information to assist generation. The common source of information for memory construction includes internal knowledge extracted from corpora and external knowledge from structured bases. For example, memory-augmented frameworks are widely adopted to project dialogue history in dialogue systems [27], while topic-to-essay generation systems integrate an external knowledge base into the generator through a dynamic memory mechanism [28]. In this paper, we follow the former approach that automatically learns useful memory information from the training corpus. The workflow is to store external memory into embedding vectors to form a memory matrix, update during training, and adopt query vectors to extract related memory as a part of the input for the generator [30].

Similar to our work, sentiment memory based auto-encoder (SMAE) [20] learns sentiment memories as style vectors to assist sentiment modification. The main differences are as follows: (1) SMAE constructs memory based on the strong assumption that all words are either style-related or irrelevant, while

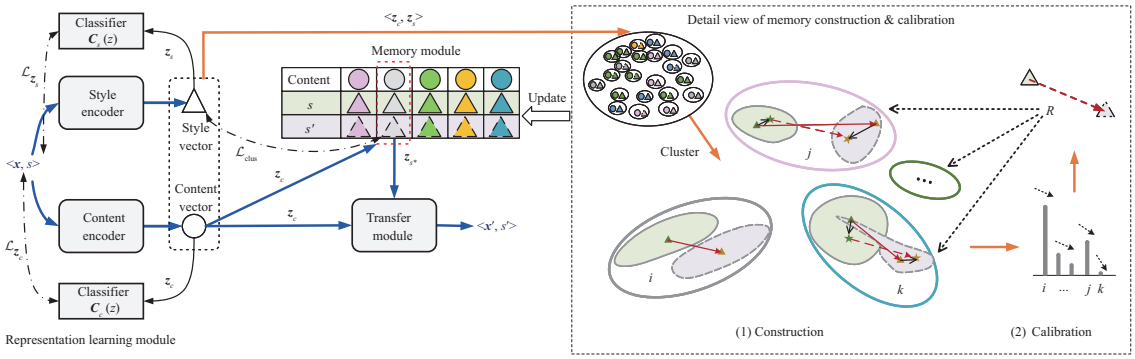

Figure 2 (Color online) The architecture of our model. Blue arrows show the transfer process, and yellow arrows illustrate memory construction and calibration processes. $\langle \boldsymbol{x}, s \rangle$ indicates the input sequence $\boldsymbol{x}$ with its style s, and $\langle \boldsymbol{x}', s' \rangle$ is the transferred output $\boldsymbol{x}'$ with desired style s'. The content vector $\boldsymbol{z}_c$ is used as a query to fetch desired style signal from memory to assist transfer. The dotted box illustrates how to memorize information by clustering, and learn relationships to calibrate memory. The dashed arrows show the training objectives of our method. We omit the reconstruction and cycle reconstruction tricks, i.e., $\mathcal{L}_{\text{self}}$ and $\mathcal{L}_{\text{cycle}}$, for simplicity.

our model does not require this assumption. In contrast, we propose a memory module to memorize content-style pairs as well as their dependencies. Hence, our model can be extended to a wide range of linguistic styles, such as sentiment, formality, and authorship. (2) Our model addresses the imbalanced sample quantity and quality problem through memory calibration, unlike SMAE. The skew distribution of training samples is likely to make the learned memory of low quality, particularly memory slots extracted from few-shot samples. Thus, we further incorporate an alignment constraint in the embedding space to augment these weak slots.

3 Methodology

3.1 Task definition

Consider a training corpus $\mathcal{X} = \{(\boldsymbol{x}_i, s_i)\}$, where each instance indicates a sentence $\boldsymbol{x}_i$ with its labeled style attribute s_i (e.g., sentiment and formality). Specifically, given a piece of sentence $\boldsymbol{x}$ carrying the original style s, TST aims to learn a transfer model $f_\theta(\boldsymbol{x}, s')$ to generate a sentence $\boldsymbol{x}'$ with a different desired attribute s' while maintaining unchanged content. For example, given the sentence "five stars, helpful staff and great service" with positive sentiment, TST aims to alter the sentiment to negative and output the transferred sentence "zero stars, rude staff and terrible service". Note that the dataset is nonparallel in our experimental setting; thus, we have no access to the ground truth sentence $\boldsymbol{x}'$.

3.2 Model architecture

As illustrated in Figure 2, our model comprises three components: a latent representation learning module, a content-style memory module, and an METM. The latent representation learning module maps text into separated content and style representations through multi-task learning. Aside from being used as the input of the transfer module, the learned latent representations are also collected for memory construction. The content-style memory module extracts useful linguistic information, including content-style correspondence characteristics, from the training corpus. The details of memory construction will be emphatically introduced in Subsections 3.3 and 3.4. The METM retrieves information from memory and merges it with the input sequence to achieve transfer.

3.2.1 *Latent representation learning module*

The latent representation learning module aims to model the underlying latent factors of content and style. We adopt two encoders, i.e., a content encoder and a style encoder, with the bidirectional gated recurrent unit (GRU) to encode the input sentence $\boldsymbol{x}$ into two latent representations, namely, the content vector $\boldsymbol{z}_c$ and style vector $\boldsymbol{z}_s$. To ensure $\boldsymbol{z}_c$ projects content information while $\boldsymbol{z}_s$ projects style information from the input sentence, we adopt multi-task learning and introduce two auxiliary tasks as supervision

for training. The common intention is that an expressive latent representation can project and recover corresponding linguistic features or properties.

For style representation learning, we conduct an additional prediction of style-oriented attributes based on the style representation $\boldsymbol{z}_s$ to facilitate modeling stylistic information. Since each sentence $\boldsymbol{x}$ is in pair with its style label s, we feed $\boldsymbol{z}_s$ into a softmax classifier $C_s(\boldsymbol{z})$ to predict the label. As with most classification tasks, the training objective is to minimize the cross-entropy loss:

$$\mathcal{L}_{\boldsymbol{z}_s} = -\mathbb{E}_{(\boldsymbol{x},s)\sim\mathcal{X}}[\log p_{C_s}(s \mid \boldsymbol{z}_s)]. \tag{1}$$

For content representation learning, we adopt a self-supervised learning method for the content-oriented constraint due to the lack of ready-made labels. Particularly, the content information is referred to as bag-of-words (BoW) features, and the auxiliary task is to recover such features based on content vector $\boldsymbol{z}_c$. Since nouns in the text usually remain unchanged during transfer, we choose nouns as candidates. In practice, our method first constructs an extra noun vocabulary $\mathbb{V}_{\text{noun}}$ by extracting all distinct nouns from the training corpus through NLTK POS tagging API. Then, given a sentence $\boldsymbol{x}$ containing N noun words $\{n_1, n_2, \ldots, n_N\}$, the BoW probability is calculated as follows: $\forall w \in \mathbb{V}_{\text{noun}}$, $p_c(w) = \frac{\sum_{i=1}^{N} \mathbb{I}\{w=n_i\}}{N}$, where $\mathbb{I}\{\cdot\}$ is the indicator function. Similarly, we adopt a softmax classifier $C_c(\boldsymbol{z})$ to predict the BoW distribution on $\boldsymbol{z}_c$. The loss function is the cross-entropy loss against the automated BoW feature $p_c(\cdot)$:

$$\mathcal{L}_{\boldsymbol{z}_c} = -\mathbb{E}_{(\boldsymbol{x},s)\sim\mathcal{X}}[\log p_{C_c}(p_c \mid \boldsymbol{z}_c)]. \tag{2}$$

In the training stage, the $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$ pairs are fed to the transfer module for rendering and also collected for the memory construction.

3.2.2 *METM*

The transfer module reads out fine-grained style signals from memory to control the style attributes of generated text. We denote the memory as $\boldsymbol{M} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K\}$, where $\boldsymbol{m}_i$ is a memory unit with learned content information as a key cell and style information as value cells. Given the content vector $\boldsymbol{z}_c$ of the sentence $\boldsymbol{x}$, the transfer module uses $\boldsymbol{z}_c$ as a query to retrieve the corresponding desired style signal $\boldsymbol{z}_{s^*}$ from memory. Next, we concatenate $\boldsymbol{z}_c$ and $\boldsymbol{z}_{s^*}$ as the initial states of the decoder. Formally,

$$p_\theta(\boldsymbol{x}' \mid \boldsymbol{x}, s') = \prod_{t=1}^{L} p_\theta(x'_t \mid \boldsymbol{z}', x'_1, \ldots, x'_{t-1}, \boldsymbol{x}), \tag{3}$$

where θ represents the parameters of the transfer module, and $\boldsymbol{z}'$ is the concatenation of $\boldsymbol{z}_c$ and $\boldsymbol{z}_{s^*}$.

3.3 Content-style memory construction

Our memory module aims to learn and store fine-grained style signals corresponding to different topics of content. Style in the text is a complex concept such that even for the same style attribute, stylistic words vary with the context. For this reason, we group the training samples into K clusters according to the semantics of content and store the common linguistics of these clusters in the content-style memory to assist transfer. Each memory unit should cover a series of transfer cases associated with a highly related topic (e.g., "cake", "steak", and "chicken" under the "food" topic) and provide corresponding style signals to generate topic-related stylistic rephrases (e.g., "delicious" for positive sentiment and "tasteless" for negative).

To this end, our memory $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_K\}$ employs a one-key-two-value structure. That is, K memory units are used, and each unit $\boldsymbol{m}_i = \langle \boldsymbol{k}_i, \boldsymbol{v}_i^s, \boldsymbol{v}_i^{s'} \rangle$ comprises a key cell $\boldsymbol{k}_i$ and two value cells $\langle \boldsymbol{v}_i^s, \boldsymbol{v}_i^{s'} \rangle$. The key cell should memorize the linguistic characteristics of a specific category of content, and the two value cells represent corresponding content-dependent style signals with respect to the candidate styles, i.e., source style s and target style s', respectively.

3.3.1 *Memory writing*

To construct a set of content-dependent style representations, we cluster training samples in the corpus into K groups according to the semantics of the content. Particularly, we collect $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$ pairs from a latent representation learning module and employ the K-Means algorithm to divide all pairs into K

clusters based on $\boldsymbol{z}_c$. Taking the i-th cluster $C_i = \{\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle\}_i$ as an example, the center $\boldsymbol{k}_i$ is referred to the average of content representations $\{\boldsymbol{z}_c\}_i$ in C_i, i.e., $\boldsymbol{k}_i = \text{AVERAGE}(\{\boldsymbol{z}_c\}_i)$, and is used as a key cell in our memory module. Accordingly, we let the average of the representations with respect to different styles (s and s') in C_i be corresponding style characteristics, i.e., the two value cells $\boldsymbol{v}_i^s$ and $\boldsymbol{v}_i^{s'}$ in our memory. Hence, the linguistic information $\langle \boldsymbol{k}_i, \boldsymbol{v}_i^s, \boldsymbol{v}_i^{s'} \rangle$ can be considered an extraction of the content-style correspondence relationship hidden in C_i. In this way, we mark the K clusters as the essential content subdivisions and consider samples in the same cluster to describe highly related objects and share similar latent style spaces. Accordingly, the key embedding $\boldsymbol{k}_i$ projects the common linguistics of the specific scope of content, and value embeddings $\langle \boldsymbol{v}_i^s, \boldsymbol{v}_i^{s'} \rangle$ retain the characteristics of candidate styles w.r.t. $\boldsymbol{k}_i$.

3.3.2 *Memory reading*

In our model, the memory read operation uses the content vector $\boldsymbol{z}_c$ of the input sequence as the retrieving head to retrieve the most relevant memory slot by measuring the similarity between $\boldsymbol{z}_c$ and the key embedding $\boldsymbol{k}_i$ of every memory slot. We design two strategies, Hard Read and Soft Read, to retrieve and construct the target style representation. Both strategies use a dot-product-based attention mechanism to loop over K memory units. Hard Read retrieves the most relevant memory slot (4) while Soft Read adopts a weighted summation over all memory slots (5).

$$\text{Read}_{\text{Hard}}(\boldsymbol{z}_c) = \left\{ \boldsymbol{v}_{i^*}^{s'} \mid i^* = \underset{i \in [1,K]}{\arg\max} (\boldsymbol{k}_i \cdot \boldsymbol{z}_c) \right\}, \tag{4}$$

$$\begin{aligned} \text{Read}_{\text{Soft}}(\boldsymbol{z}_c) &= \sum_{i=1}^{K} \alpha_i \boldsymbol{v}_i^{s'}, \\ \alpha_i &= \text{softmax}(\boldsymbol{k}_i \cdot \boldsymbol{z}_c), \end{aligned} \tag{5}$$

where $\boldsymbol{v}_i^{s'}$ is the representation of target style s' memoried in the unit $\boldsymbol{m}_i = \langle \boldsymbol{k}_i, \boldsymbol{v}_i^s, \boldsymbol{v}_i^{s'} \rangle$. We retrieve the desired style representation, i.e., $\boldsymbol{z}_{s^*}$ in Subsection 3.2.2, from memory and feed it to the transfer module.

3.3.3 *Iterative update*

Since the memory update requires a number of time steps accumulation for $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$ pairs, it cannot synchronize with the generative model training. Therefore, we divide the entire training into two stages, memory update and generative model training, and then train the two stages alternately.

At the generative model training stage, we fix the information stored in the memory and use it as a standard to participate in the transfer phase. The generative model retrieves the desired style signals from memory for the style control of the output, thereby optimizing itself. At the memory update stage, the generative model that has been optimized through epoch training creates more precise $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$ pairs of the training corpus, which contributes to improving clustering quality. We conduct clustering on the new collection and write the results into the memory for an update.

The two stages interact to achieve overall optimization iteratively. During training, the latent representation learning module trains to optimize itself and generate a collection of more accurate and expressive $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$ pairs. Therefore, our memory is also updated iteratively to adapt to new $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$ pairs regularly. In practice, the two stages interchange once per epoch; thus, we update the memory each time the generative model finishes epoch training.

Intuitively, the performance of our memory is highly correlated with the quality of clustering. For this reason, we also heed the potential error deviation caused by outliers or the skew distribution of samples in the clustering results, particularly for the few-shot cluster. To address such issues, we further introduce a way to calibrate style signals stored in the value cells after updating the memory.

3.4 Memory calibration

To alleviate the error deviation of latent style space caused by uneven qualities of clusters, we take the insights from TransE [15] and propose a memory calibration method. Our motivation is based on the assumption that the relationship between styles follows a similar distribution across all memory units.

Therefore, our method first learns a standard relationship representation and uses it as a shared criterion to calibrate the value embeddings of memory.

Particularly, for the two value cells ($\boldsymbol{v}_i^s$ and $\boldsymbol{v}_i^{s'}$), the relationship is defined as a translation of value embeddings:

$$\boldsymbol{r}_i = \boldsymbol{v}_i^s - \boldsymbol{v}_i^{s'}, \tag{6}$$

where $\boldsymbol{r}_i$ indicates the relationship between $\boldsymbol{v}_i^s$ and $\boldsymbol{v}_i^{s'}$.

Then, we construct a standard relation representation by taking the weighted average of all relationship vectors of K memory units by

$$\boldsymbol{R} = \sum_{i=1}^{K} \alpha_i \boldsymbol{r}_i, \tag{7}$$

where α_i is the weight of $\boldsymbol{r}_i$, which is determined by the quality of style distribution in the i-th cluster.

We consider a cluster with good quality if the groups of different styles in the cluster are well apart from each other and clearly distinguished. That is, we expect a sample with style s to be near samples sharing the same style while keeping away from other samples with different style s' as much as possible. The silhouette coefficient algorithm [31], which measures the separation between clusters, is adopted to evaluate whether the different groups of style distributions are well separated, where a higher score means a better cluster quality. We calculate the silhouette coefficient scores over all clusters as the distribution weights. Specifically,

$$\alpha_i = \frac{\exp(e_i)}{\sum_{i=0}^{K} \exp(e_i)}, \tag{8}$$

where e_i is the silhouette coefficient score for cluster i.

Finally, we calibrate value embeddings over the memory module by decreasing the deviation between the relation vector of each cluster and the standard relation vector $\boldsymbol{R}$.

$$\boldsymbol{r}_i^{\text{bias}} = \boldsymbol{R} - \boldsymbol{r}_i,$$

$$\widehat{\boldsymbol{v}_i^s} = \boldsymbol{v}_i^s + \lambda \boldsymbol{r}_i^{\text{bias}},$$

$$\widehat{\boldsymbol{v}_i^{s'}} = \boldsymbol{v}_i^{s'} + \lambda \boldsymbol{r}_i^{\text{bias}},$$

where $\boldsymbol{r}_i^{\text{bias}}$ indicates the relationship bias on the embedding space and λ is a hyperparameter for controlling modification strength. $\widehat{\boldsymbol{v}_i^s}$ and $\widehat{\boldsymbol{v}_i^{s'}}$ denote optimized latent style representations in the value cells.

3.5 Unsupervised training

To effectively support training on nonparallel corpora, we define the following losses to provide supervision indirectly.

3.5.1 *Reconstruction loss*

For the transfer case, in which the source style is identical to the desired style, i.e., $s' = s$, our model should reconstruct the original sentence:

$$\mathcal{L}_{\text{self}} = -\mathbb{E}_{(\boldsymbol{x},s)\sim\mathcal{X}}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{x}, s)]. \tag{9}$$

3.5.2 *Cycle reconstruction loss*

For the case $s' \neq s$, we first generate the transferred sentence $\boldsymbol{x}'$, and in turn switch the transfer direction to rephrase $\boldsymbol{x}'$ back to $\boldsymbol{x}$:

$$\mathcal{L}_{\text{cycle}} = -\mathbb{E}_{(\boldsymbol{x},s)\sim\mathcal{X}}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{x}', s)]. \tag{10}$$

Table 1 Dataset statistics

Dataset	Yelp		Amazon		GYAFC	
	Positive	Negative	Positive	Negative	Formal	Informal
Train	270k	180k	277k	278k	52k	52k
Dev.	2000	2000	985	1015	2247	2788
Test	500	500	500	500	1019	1332
Ref.	500	500	500	500	1019	1332
Avg.Len.	8.9		14.9		12.7	

3.5.3 *Style transfer loss*

Since the ground truth $\boldsymbol{x}'$ is unavailable, we apply an adversarial discriminator module that distinguishes the style of text to provide style supervision. Particularly, the discriminator is trained to identify the fake transferred sentence from the generator. In contrast, the training objective of the generator is to fool the discriminator, i.e., maximize the probability of s' when given the transferred sentence $\boldsymbol{x}'$:

$$\mathcal{L}_{\text{style}} = -\mathbb{E}_{(\boldsymbol{x},s)\sim\mathcal{X}}[\log p_C(s' \mid \boldsymbol{x}')]. \tag{11}$$

The sampling process or greedy decoding while generating tokens prevents the gradients from propagating, so we follow Dai et al. [11] and adopt the continuous decoding algorithm for token generation. Specifically, $\boldsymbol{x}'$ is generated by the weighted sum embedding of softmax distribution on the embedding matrix rather than selecting the maximum probability token each time step.

3.5.4 *Cluster loss*

Given a pair $\langle \boldsymbol{z}_c, \boldsymbol{z}_s \rangle$, we use $\boldsymbol{z}_c$ as a query to retrieve a set of style representations from memory, including the one associated with original style, denoted as $\boldsymbol{z}_{\hat{s}}$. We expect $\boldsymbol{z}_s$ to be near $\boldsymbol{z}_{\hat{s}}$ in the latent space because they represent similar content-dependent style signals.

$$\mathcal{L}_{\text{clus}} = \mathbb{E}_{(\boldsymbol{x},s)\sim\mathcal{X}} \|\boldsymbol{z}_s - \boldsymbol{z}_{\hat{s}}\|_2^2. \tag{12}$$

3.5.5 *Overall loss*

Recalling $\mathcal{L}_{\boldsymbol{z}_s}$ (1) and $\mathcal{L}_{\boldsymbol{z}_c}$ (2) in Subsection 3.2.1, our overall loss function is a synthesis of six parts

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{self}} + \lambda_2\mathcal{L}_{\text{cycle}} + \lambda_3\mathcal{L}_{\text{style}} + \lambda_4\mathcal{L}_{\boldsymbol{z}_s} + \lambda_5\mathcal{L}_{\boldsymbol{z}_c} + \lambda_6\mathcal{L}_{\text{clus}}, \tag{13}$$

where λ_i is the balancing parameter.

4 Experiment

4.1 Datasets

In this paper, we evaluate our method on two typical TST tasks: sentiment modification [11, 12, 19] on the Yelp and Amazon corpora and formality transfer [25,26] on the GYAFC corpus. For fair comparisons, we adopt the common preprocessing methods proposed by Dai et al. [11] and Xiao et al. [26] for data construction. Table 1 illustrates the detailed dataset statistics. The first two datasets comprise positive and negative reviews for Yelp restaurants and Amazon products, respectively. The GYAFC dataset, i.e., Grammarly's Yahoo Answers Formality Corpus provided by Rao et al. [32], contains formal and informal sentences in two domains, namely, Entertainment & Music and Family & Relationships. Following previous work [25, 26], we choose the latter domain and use it in an unpaired setting during training. Moreover, all three datasets release human-written references of the test set for direct evaluation.

4.2 Baselines

We compare two versions of our proposed METM, i.e., METM-S and METM-H represent soft reading and hard reading strategies, respectively, with several competitive baselines covering disentanglement-based and attention-based paradigms.

We choose the following disentanglement-based competitor models that learn style-independent content representation and fuse them with desired style control variables for transfer.

• **CrossAlign** [6]**.** CrossAlign learns style-independent content representation and fuses it with desired style embedding to generate transferred sentences.

• **MultiDec** [17]**.** MultiDec adopts an adversarial network to guarantee disentanglement and then feeds content representation to multiple style-specific generators without additional style control variables.

• **DelRetGen** [2]**.** DelRetGen extracts a sentence template by removing stylistic phrases and applies a generative model to fuse it with retrieved phrases carrying the desired attribute for transfer.

• **SMAE** [20]**.** SMAE deletes sentimental words through a self-attention based classifier and learns sentiment memories to adapt different contexts for sentiment modification.

• **Revision** [33]**.** Revision revises the original sentences in a continuous space by gradient optimization in the inference stage to achieve transfer.

We consider the following representative methods for the attention-based paradigm, which adopt the Seq2Seq structure to obtain entangled representation without disentanglement.

• **DualRL** [19]**.** DualRL is a dual reinforcement learning algorithm that treats style transfer as a dual task without separating content and style.

• **StyTrans** [11]**.** StyTrans employs Transformer architecture to obtain entangled representation. It injects the stylistic property as an extra embedding added to the entangled embedding.

• **IMaT** [18]**.** IMaT constructs a pseudo-parallel corpus by iteratively matching and refining semantically similar sentences and applies a Seq2Seq-based transfer model to learn attribute transfer.

• **PFST** [24]**.** PFST introduces a probabilistic generative paradigm and achieves transfer by modeling a transduction distribution on latent space.

• **StyIns** [12]**.** StyIns adopts the attention-based structure and learns latent style space from multiple instances.

• **DIRR** [25]**.** DIRR proposes a reward-based training algorithm and uses a semantic similarity metric to enhance content preservation.

• **TSST** [26]**.** TSST extracts stylistic properties from retrieval-based top-k relevant instances to provide a strong style signal.

4.3 Automatic evaluation

Given the main characteristics of the transfer task, we follow the common practice of considering three aspects, i.e., style transfer strength, content preservation, and language fluency, in the automatic evaluation metrics [12, 19, 26]. In addition, we further heed the lexical diversity of the output to judge whether the candidate systems can generate varied style-related phrases or simply prefer similar generic context-independent phrases.

• **Style transfer strength.** To measure style transfer strength quantitatively, we calculate the style accuracy (Acc.) of transferred sentences through a fine-tuned BERT-based [34] classifier. The accuracies of well-trained classifiers reach 98% on Yelp, 89% on Amazon, and 90% on GYAFC.

• **Content preservation.** Following the standard practice, we report the self-BLEU (s-BLEU) metric between the input and the output of transfer systems, where a higher score indicates better content preservation to some extent. Moreover, we also calculate the ref-BLEU (r-BLEU) score for direct evaluation by comparing the output with human-written references. To ensure the fairness of the comparison, we adopt the NLTK BLEU scoring function [35] for all BLEU calculations, similar to Yi et al. [12] and Xiao et al. [26].

• **Language fluency.** The measurement of fluency is the perplexity (PPL) of generated sentences. Following previous work [11, 12, 26], we adopt KenLM [36] to train 5-gram language models on three datasets to calculate PPL.

• **Diversity.** We report the Distinct-1 (Dist-1) and Distinct-2 (Dist-2) scores [37] that calculate the proportion of distinct unigrams/bigrams in the transferred results to indicate the diversity.

• **GM.** Intuitively, a trade-off is made between modifying the style and preserving the content [12,14]. For this reason, we further apply a geometric mean of transfer accuracy, self-BLEU, ref-BLEU, and $\frac{1}{\log \mathrm{PPL}}$, for overall quality evaluation, denoted as GM [26]. Note that we omit Distinct-1 and Distinct-2 to calculate GM, which is unchanged from previous studies for a fair comparison.

Table 2 Automatic evaluation results on the sentiment modification task[a)]

Type	Method	Yelp							Amazon						
		Acc.↑	r-BLEU↑	s-BLEU↑	PPL↓	GM↑	Dist-1↑	Dist-2↑	Acc.↑	r-BLEU↑	s-BLEU↑	PPL↓	GM↑	Dist-1↑	Dist-2↑
Disen.	CrossAlign	78.7	8.11	16.65	66	7.10	0.088	0.531	69.6	2.02	2.84	95	3.06	0.119	0.475
	MultiDec	45.4	15.07	40.07	188	8.51	0.111	0.701	66.5	6.99	16.34	88	6.42	0.065	0.513
	DelRetGen	88.1	16.66	36.75	100	10.40	0.154	0.617	52.4	29.14	53.31	85	11.63	0.151	0.583
	SMAE	76.6	15.24	43.05	65	10.47	0.172	0.683	70.4	15.44	45.43	92	10.22	0.164	0.627
	Revision	90.6	7.93	13.23	21	7.47	0.111	0.432	**80.7**	13.99	20.07	**38**	8.88	0.127	0.497
Attn.	DualRL	87.9	28.77	58.90	105	13.38	0.142	0.643	62.4	24.18	49.17	124	11.14	0.152	0.675
	StyTrans	86.0	27.32	59.46	154	12.91	0.168	0.691	65.5	23.58	61.92	208	11.57	0.142	0.636
	IMaT	**93.9**	11.26	16.92	**14**	9.07	0.120	0.480	72.4	12.65	33.95	58	9.35	0.140	0.633
	PFST	84.6	23.72	48.90	67	12.36	0.153	0.628	61.8	34.81	66.92	87	13.40	0.170	0.677
	StyIns	90.9	26.09	53.10	110	12.79	0.147	0.664	62.3	22.42	43.39	115	10.63	0.128	0.425
	DIRR	91.7	28.54	58.98	144	13.28	0.158	0.636	58.8	32.80	66.38	110	12.85	0.173	0.662
	TSST	91.8	28.89	59.34	108	13.54	0.141	0.655	59.4	41.59	69.95	130	13.73	0.171	0.694
Ours	METM-S	92.3	28.62	61.42	112	13.62	0.164	0.667	66.0	40.51	**70.02**	126	14.03	0.181	0.696
	METM-H	91.1	**29.34***	**63.49***	113	**13.76***	**0.174**	**0.701**	62.2	**43.68***	68.77	121	**14.05***	**0.189***	**0.707***

a) Bold cells indicate the best performances. We conduct hypothesis testing and the results marked with star are statistically significant with $p < 0.05$ under t-test.

4.4 Human evaluation

Because of the time and economical consumption of human annotation, we choose the four most competitive methods with the highest GM metric and our METM models as candidates. In practice, we randomly sample 100 transfer cases (50 cases for each style) from the output of each transfer system (1800 cases in total on all three datasets). Each case contains the original sentence, target style attribute, and generated sentence. We then distribute these samples to three annotators for giving a score range from 1 (the worst) to 5 (the best) regarding three common criteria: style transfer accuracy (Style), content preservation (Content), and Fluency. Similar to Distinct-n metrics for lexical diversity in automatic evaluation, these annotators also need to evaluate the overall linguistic diversity (Diversity) of stylistic expressions from different systems. Note that the diversity metric focuses on the overall lexical diversity rather than a property for a single case. In practice, we divide the sampled cases into groups of 10 and then ask the annotator to calculate the number of unique style expressions (i.e., ranging from 1 to 10) in each group. The evaluation is conducted in a strictly random and blind fashion to avoid human bias. Considering the workload and difficulty, the expected annotation time is 18 h (100 sentences per hour). Therefore, each annotator is compensated with 1800 Chinese Yuan (CNY), and the hourly pay is CNY100, which exceeds the Chinese statutory minimum wage.

4.5 Implementation details

In this paper, we employ a one-layer bidirectional GRU for the style and content encoders and a one-layer unidirectional GRU with the attention mechanism for the decoder. We use 300-dim pretrained GloVe [38] as word embeddings shared by encoders and the decoder. Correspondingly, the hidden state size is also set to 300 for encoders and the decoder. Because of the powerful ability of the Transformer, the discriminator adopts a four-layer Transformer with four-way multi-head attention. In the training stage, we set the memory size to 50 and employ the Adam optimizer [39] with an initial learning rate of 0.0001 and a gradient clipping of 5. In (13), we follow Xiao et al. [26] to set λ_1, λ_2, and λ_3 to 1 for balancing style transfer strength and content preservation, while tuning $\lambda_4 = 0.5$, $\lambda_5 = 0.5$, and $\lambda_6 = 0.25$ through manual search optimization.

4.6 Results and analysis

4.6.1 *Automatic evaluation results*

Table 2 shows the automatic evaluation results of the sentiment modification task. Generally, disentanglement-based methods achieve comparable or even better transfer accuracy and PPL but perform much worse in content preservation, i.e., ref-BLEU and self-BLEU scores. For instance, the Revision model achieves promising or near-best performances on transfer accuracy and language fluency, particularly

Table 3 Automatic evaluation results on the formality transfer task[a)]

Type	Method	GYAFC						
		Acc.↑	r-BLEU↑	s-BLEU↑	PPL↓	GM↑	Dist-1↑	Dist-2↑
Disen.	CrossAlign	61.6	3.25	2.21	**37**	3.33	0.012	0.081
	MultiDec	24.5	11.95	16.08	151	5.53	0.021	0.320
	DelRetGen	58.2	21.88	31.57	103	9.65	0.020	0.246
	SMAE	59.5	23.46	43.15	87	10.78	0.066	0.451
	Revision	39.6	20.83	27.64	66	8.59	0.025	0.096
Attn.	DualRL	55.5	43.69	52.80	159	12.61	0.063	0.473
	StyTrans	60.3	43.95	61.15	168	13.34	0.067	0.469
	IMaT	63.8	4.18	53.77	46	7.82	0.072	0.434
	PFST	63.9	21.53	28.68	40	10.17	0.066	0.342
	StyIns	69.9	47.80	61.87	140	14.30	0.066	0.473
	DIRR	71.8	46.37	59.97	145	14.15	0.061	0.417
	TSST	74.1	50.49	63.70	103	15.06	0.061	0.468
Ours	METM-S	**75.6***	49.74	61.64	87	15.09	0.069	**0.478**
	METM-H	74.0	**52.05***	**64.08***	99	**15.22***	**0.072**	0.477

a) Bold cells indicate the best performances. We conduct hypothesis testing and the results marked with star are statistically significant with $p < 0.05$ under t-test.

on Amazon, reaching 80.7 and 38 in terms of accuracy and PPL metrics, respectively, but suffers from a substantial performance decrease on content preservation across all datasets. These results illustrate that disentanglement-based methods tend to generate fluent sentences with target stylistic properties but are irrelevant to original content semantics. We attribute this attribute to the intractability of disentanglement, which damages the content information when separating the style from the text.

In contrast, attention-based methods avoid explicit disentanglement and thus better preserve content information. Almost all attention-based models except IMaT marginally outperform disentanglement-based methods regarding ref-BLEU and self-BLEU metrics. For example, TSST gains a remarkable improvement of nearly 10 points on all BLEU scores compared to well-performed disentanglement-based models. IMaT is likely an exception because the method for constructing a pseudo-parallel corpus implicitly requires the assumption of disentanglement, thereby conforming with a similar pattern as the disentanglement-based paradigm. Despite enhanced content retainment, attention-based methods perform slightly worse for the transfer success ratio, and in particular, a show nontrivial drop on Amazon. Representative methods, including DualRL, StyTrans, and PFST, consistently underperform regarding transfer accuracy compared to the SOTA obtained by the disentanglement-based line, which verifies that taking simple style embedding is insufficient for guiding transfer.

For the formality transfer task on the GYAFC dataset, Table 3 reports the experimental results of our method and competing methods. Similarly, attention-based methods generally surpass disentanglement-based methods regarding content preservation. Nevertheless, the comparison of transfer strength (Acc.) shows a slight difference from the previous observation. In the previous sentiment modification task, the SOTA disentanglement-based methods achieve comparable or even higher accuracy compared to the attention-based line. However, as they migrate to the formality transfer task, disentanglement-based methods suffer from a dramatic decrease in transfer accuracy and are generally inferior to the attention-based line. For example, Revision achieves promising performance in terms of accuracy on the Yelp and Amazon datasets, reaching 90.6 and 80.7, respectively, but sharply drops to only 39.6 on the GYAFC dataset. We deduce that this substantial difference is obtained because the disentanglement of style attributes and content information is more challenging in the formality transfer task than sentiment modification. This phenomenon indicates that assuming disentanglement may limit the application scope in practice.

Moreover, the diversity metrics, including Distinct-1 and Distinct-2, vary substantially between different approaches. Despite the acceptable transfer accuracy, these baselines that adopt simple style embedding as the controllable signal, e.g., CrossAlign, Revision, and IMaT, consistently perform unsatisfactorily in lexical diversity. The results demonstrate that these models only generate universal style-related phrases independent of the input and prove our claim that solo embedding is insufficient for representing linguistic style. In comparison, representing style in a fine-grained form provides noticeable gains in diversity metrics. For instance, TSST enhances style signals by incorporating retrieval samples

Table 4 Human evaluation results[a)]

Method	Yelp				Amazon				GYAFC			
	Style	Content	Fluency	Diversity	Style	Content	Fluency	Diversity	Style	Content	Fluency	Diversity
DualRL	3.40	3.75	**4.18**	5.13	3.09	2.96	3.47	6.87	1.97	3.07	3.20	5.17
StyIns	3.47	3.74	4.14	6.67	2.67	2.73	3.82	6.20	2.26	2.98	2.31	5.83
DIRR	3.69	3.87	3.79	5.37	2.26	3.32	**4.04**	7.07	2.78	3.41	3.51	4.86
TSST	3.81	4.05	4.07	6.17	2.43	3.74	3.79	7.83	2.73	3.47	3.77	5.37
METM-S	**3.82**	4.09	4.13	7.50	3.02	3.37	3.85	8.10	**2.96**	3.27	**4.05**	5.83
METM-H	3.78	**4.18**	4.02	**8.07**	**3.14**	**3.75**	3.98	**8.23**	2.78	**3.55**	3.84	**6.07**

a) The best results are shown in bold. The Krippen-dorff's alpha of human rating is 0.7727, indicating moderate inter-annotator agreement.

and achieves considerable improvement across all three datasets, indicating the advantage of fine-grained style representation in expressing diverse stylistic phrases.

Our model consistently outperforms benchmark models on most metrics, and particularly encouraging improvements are obtained regarding the ref-BLEU score, indicating that the output of our model has more similar context semantics and style attributes to manual references. Intuitively, a trade-off is made between the transfer accuracy and content preservation criteria [12,14]. As shown, our proposed METM model achieves a better balance of transfer accuracy and content preservation. The highest GM metrics on all three datasets show the superiority of our model in not only altering style but also retaining content. Crucially, both versions of our models achieve remarkable improvements in lexical diversity, verifying that fine-grained content-dependent style signals confer an important advantage in expressing diverse stylistic attributes. We also find that our method achieves the most remarkable overall performance improvement on the Amazon dataset. To explain this phenomenon, we count the number of unique nouns in each dataset (Yelp: 7750, Amazon: 40505, GYAFC: 13285), showing that product reviews on Amazon involve the broadest range of objects. This finding supports the idea that our memory module affects the stylistic generation, particularly when adapting to various topics of content. For two versions of our models, METM-H slightly outperforms METM-S on most quantitative metrics, indicating that the hard reading strategy is slightly better than soft reading for retrieving the desired style representation.

4.6.2 *Human evaluation results*

Table 4 shows the human evaluation results, which are highly consistent with automatic evaluation results regarding transfer accuracy, content preservation, and diversity. Our proposed METM model achieves comparable fluency compared to the most well-performed models under manual evaluation, proving its capacity for generating fluent transferred sentences with desired style properties. This result indicates that a moderate PPL metric (not necessarily overly low PPL) meets the fluency requirement, consistent with previous studies [12,24]. Additionally, our model marginally outperforms all four competitor models in terms of diversity scores, verifying its superiority in generating diverse stylistic phrases that vary with the context. Given that stylistic expressions vary substantially under different contexts, our proposed content-style memory module provides fine-grained content-dependent style signals rather than taking a simple embedding to control transfer. The more expressive and flexible style representations make the model avoid generating tired content-irrelevant stylistic phrases and confer a substantial advantage in improving lexical diversity.

By modeling an expressive latent style space through transductive learning, TSST achieves competitive performance in all four indicators. Even so, our model outperforms TSST under most metrics. We attribute this supremacy to a superiority of style representation construction. Instead of only retrieving several samples to distill the desired style representation, our method integrates all training samples through clustering to learn a more expressive and flexible latent style space. In addition, the calibration operation on stylistic properties also potentially provides further performance gains.

4.6.3 *Ablation study*

To study the effects of the critical components of our method, we conduct an ablation study of our method, the results of which are presented in Table 5 (for only the Yelp dataset because of limited space). We first investigate the impact of the memory module by presenting two variants. The first one (denoted as (-)Memory) replaces the memory module with typical style embeddings. The variant consistently under-

Table 5 Model ablation study result on Yelp dataset

Model	Acc.	r-BLEU	s-BLEU	PPL	GM	Dist-1	Dist-2
METM-H	91.1	29.34	63.49	113	13.76	0.174	0.701
(-)Memory	91.0	26.88	55.63	115	13.01	0.158	0.626
(-)Calibrate	91.3	28.41	60.72	108	13.54	0.163	0.638
(-)$\mathcal{L}_{\text{self}}$	98.2	5.17	10.34	797	5.29	0.047	0.101
(-)$\mathcal{L}_{\text{cycle}}$	94.0	23.25	46.22	161	11.87	0.147	0.603
(-)$\mathcal{L}_{\text{style}}$	2.0	32.23	98.69	117	6.05	0.170	0.639
(-)$\mathcal{L}_{\boldsymbol{z}_s}$	92.2	27.57	59.69	111	13.40	0.156	0.634
(-)$\mathcal{L}_{\boldsymbol{z}_c}$	90.5	28.55	63.80	116	13.65	0.161	0.633
(-)$\mathcal{L}_{\text{clus}}$	90.7	28.71	61.41	123	13.50	0.162	0.700

Table 6 The performance of METM-H with different memory sizes K[a)]

METM-H	Acc.	r-BLEU	s-BLEU	PPL	GM	Dist-1	Dist-2
$K = 1$	91.3	26.42	54.48	109	12.94	0.147	0.623
$K = 10$	89.9	28.35	61.29	112	13.49	0.166	0.651
$K = 50$	91.1	**29.34**	**63.49**	113	**13.76**	0.174	**0.701**
$K = 100$	**91.9**	28.85	60.91	**109**	13.62	**0.176**	0.679
$K = 500$	91.1	28.60	61.30	113	13.56	0.163	0.642

a) The best results are shown in bold.

performs our model across all aspects and particularly suffers from an extremely substantial performance decrease in content preservation, i.e., the ref-BLEU and self-BLEU scores. This result illustrates that only simple style embedding as guidance signals is likely to make the transferred sentence not irrelevant to the original content to some extent. Instead, fine-grained content-dependent style expressions in our memories enable the model to generate style-related phrases consistent with the original context, thus contributing to retaining content information. In contrast, the other variant (denoted as (-)Calibrate) only removes the memory calibration part, showing that calibration further boosts overall performance. We conjecture that this boost is due to the style representations on cells with few samples and cells with many outliers introducing noise, while calibration works on these two types of cells and potentially provides further performance gains. To better understand the role of varied losses in unsupervised training, we further disable all six losses in (13) by turns. $\mathcal{L}_{\text{self}}$, $\mathcal{L}_{\text{cycle}}$, and $\mathcal{L}_{\text{style}}$ are the fundamental signals for unsupervised training, and the former two mainly promote retaining content through maximum likelihood estimation, while the latter signal facilitates style modification. The removal of $\mathcal{L}_{\boldsymbol{z}_s}$ leads to a drop in transfer accuracy; thus, the auxiliary style classification task on $\boldsymbol{z}_s$ helps obtain discriminative style representation. Conversely, $\mathcal{L}_{\boldsymbol{z}_c}$ mainly contributes to enhancing content preservation. $\mathcal{L}_{\text{clus}}$ can bring further performance improvements. In short, all six losses interact to achieve SOTA performances on unsupervised TST.

4.6.4 *Memory analysis*

To investigate the effect of different memory capacities, we conduct comparison experiments on METM-H with a range of memory sizes K and report the results in Table 6. Our memory with a setting of $K = 1$ is similar to typical individual-style embeddings; correspondingly, the performances are comparable to the variant without the memory module. A small memory ($K \leqslant 10$) is insufficient for memorizing various linguistic characteristics, bringing only slight improvements. Generally, increasing K promotes learning more expressive style representation, thereby achieving better performance. However, overly high memory sizes ($K \geqslant 100$) perform worse due to the raised error derivation in few-shot clusters. Moreover, the required resources and training time increase substantially with excessive K. In conclusion, moderate K is appropriate, and we set $K = 50$ by comparison.

Figure 3 examines how clustering attributes (i.e., average silhouette coefficient scores and the Euclidean distance between relationship vectors $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{|K|}\}$) change during training. The increasing silhouette coefficient scores show that groups with different styles are becoming increasingly more discriminative, while the decrease in average distance verifies that the relationship between different styles has a good similarity across various content topics. Crucially, the calibration operation helps learn more discriminative and expressive latent style space.

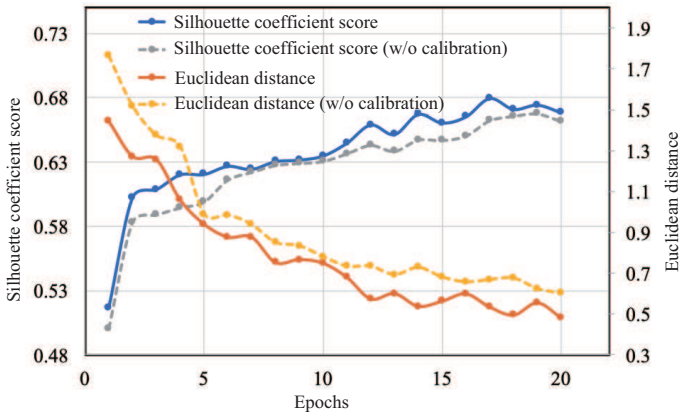

Figure 3 (Color online) The variation of clustering attributes during training. A higher silhouette coefficient score indicates more discriminative latent style space. A lower euclidean distance means a higher similarity.

Figure 4 (Color online) Visualization of samples from different memory units. (a) Unit associated with food; (b) unit associated with people.

Furthermore, we pick up the samples divided into the same cluster (memory unit) and visualize their stylistic attributes through a word cloud (the two units in Figure 4). We find that each memory unit shares a closely related topic and learns varied stylistic phrases with respect to specific content. Specifically, the left unit is related to the food topic and supplies the corresponding expressions, while the right unit mainly focuses on the properties associated with the person topic. In conclusion, the memory units cluster similar cases together and leverage fine-grained style signals to control transfer.

4.6.5 *Case study*

To intuitively compare the characteristics of different models, Table 7 shows sampled output sentences from our model and four of the most competitive baselines. In the sentiment modification case, almost all baselines can alter the style of the input sentences to the expected stylistic attributes. Even so, several baselines simply add the adverb not to achieve sentimental transformation, e.g., change great to not great in the sentimental positive-to-negative transfer direction, resulting in a relatively monotonic pattern in the transferred sentences. Moreover, we can observe that the baselines tend to dismiss part of the original semantics more or less, particularly when there are more than two described objects, e.g., their tone and customer service, in the input sentences. In comparison, our model can achieve successful transfer in generating more vivid sentimental phrases and better retainment of original content. For the formality transfer case, several baselines do not conduct effective rephrasing and, similarly, fail to remain with the content information unchanged. In contrast, our model confers a substantial advantage in preserving content under the premise of successfully altering formality. For example, as an alternative to the original phrase when ur ready, the generated phrase when you are ready correctly improves lexical formality and better preserves semantics compared to when ready.

Table 7 Case study on sentiment modification and formality transfer tasks[a)]

Sentiment modification		
Model	Positive to negative	Negative to positive
Input	i love this place, the service is always great!	always rude in their tone and always have shitty customer service!
DualRL	i **hate** this place, the service is **not** great!	always friendly in their best price and always have loved customer service!
StyIns	i do not know, the service is not even filtered cove.	always **nice** in their tone and always have **wonderful** customer service!
DIRR	**hate** this place, service was bad.	such **nice** customer service, they listen to anyones concerns and assist them with it.
TSST	i **hate** this place, the service is **not** great!	always friendly in their service and always have **great** customer service!
Ours	i **hate** this place, the service is always **terrible**!	always **friendly** in their tone and always have **great** customer service!
Formality transfer		
Model	Informal to formal	Formal to informal
Input	it all depends on when ur ready.	the two of you should kiss when you are ready.
	stay 100 miles away from this guy.	yes i am a male, therefore i do not really have to pay.
DualRL	it all depends on when ready.	the two of you should kiss when you are ready!!!!
	stay miles away from 100 guy.	yes i am a male er.
StyIns	it all depends on when your ready.	**2** of you should kiss when you are ready.
	stay an miles away from this **man**.	yes i am a male girl lol i do **n't** really have to pay.
DIRR	it all depends on when **you are** ready.	the **2** of you dont kiss when you are ready.
	stay away from this guy.	yes i **'m** a girl freind therefore i do **n't** really have to pay.
TSST	it all depends on when your ready.	and **2** of you should kiss when you are ready.
	stay 100 miles away from this **man**.	yes i **'m** a kid, therefore i do **n't** really have to pay.
Ours	it all depends on when **you are** ready.	the **2** of you kiss when **u are** ready.
	stay 100 miles away from this **man**.	yes i **'m** a **girl**, **yeah** i do **n't** really have to pay.

a) Phrases in bold mean successful transfer in style.

5 Conclusion

In this paper, we propose a memory-enhanced method with dynamic style learning for TST. Instead of representing style with individual embedding, we construct a content-style memory to learn a more expressive and flexible latent style space. Our model clusters the training corpora and extracts common linguistic features for style transfer. Moreover, we introduce a way to further calibrate memory by projecting the relationship between styles. Thereby, our model can provide dynamic style signals with respect to content, which contributes to generating diverse and informative sentences. Experimental results on three datasets verify the superiority of our model in terms of style control and content preservation.

6 Ethical considerations

Our work focuses on TST, which controls the stylistic properties of generated text while retaining content semantics. Such methods have a broad impact in the field of controllable natural language generation [40] and can provide strong support for potential real-world applications, e.g., stylized response generation [41], stylistic summarization [3], text simplification [4], and offensive language transfer [42, 43]. Nonetheless, as with all TST methods, our method can also potentially be used maliciously with concealed intentions, including possible content manipulation and forgery issues, e.g., fake review generation. For this reason, we restrict the proposed method to academic use only, and it must be coupled with strict misrepresentation, offensiveness, and bias checks. Furthermore, with increasing attention to shared ethical issues in text generation models, we encourage future studies to address such cases.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62106275).

References

1 Xu J, Sun X, Zeng Q, et al. Unpaired sentiment-to-sentiment translation: a cycled reinforcement learning approach. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 979–988
2 Li J, Jia R, He H, et al. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Louisiana, 2018. 1865–1874
3 Fan A, Grangier D, Auli M. Controllable abstractive summarization. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Melbourne, 2018. 45–54
4 Cao Y, Shui R, Pan L, et al. Expertise style transfer: a new task towards better communication between experts and laymen. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020. 1061–1071

5 Hu Z, Yang Z, Liang X, et al. Toward controlled generation of text. In: Proceedings of International Conference on Machine Learning, Sydney, 2017. 1587–1596
6 Shen T, Lei T, Barzilay R, et al. Style transfer from non-parallel text by cross-alignment. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 2017. 6833–6844
7 Zhang Z, Ren S, Liu S, et al. Style transfer as unsupervised machine translation. 2018. ArXiv:1808.07894
8 John V, Mou L, Bahuleyan H, et al. Disentangled representation learning for non-parallel text style transfer. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 424–434
9 Lee J. Stable style transformer: delete and generate approach with encoder-decoder for text style transfer. In: Proceedings of the 13th International Conference on Natural Language Generation, Dublin, 2020. 195–204
10 Tian Y, Hu Z, Yu Z. Structured content preservation for unsupervised text style transfer. 2018. ArXiv:1810.06526
11 Dai N, Liang J, Qiu X, et al. Style transformer: unpaired text style transfer without disentangled latent representation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 5997–6007
12 Yi X, Liu Z, Li W, et al. Text style transfer via learning style instance supported latent space. In: Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, 2021. 3801–3807
13 Lample G, Subramanian S, Smith E, et al. Multiple-attribute text rewriting. In: Proceedings of the International Conference on Learning Representations, New Orleans, 2018
14 Jin D, Jin Z, Hu Z, et al. Deep learning for text style transfer: a survey. 2021. ArXiv:2011.00416
15 Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Nevada, 2013. 2787–2795
16 Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 2414–2423
17 Fu Z, Tan X, Peng N, et al. Style transfer in text: exploration and evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence, Louisiana, 2018. 663–670
18 Jin Z, Jin D, Mueller J, et al. IMaT: unsupervised text attribute transfer via iterative matching and translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, 2019. 3097–3109
19 Luo F, Li P, Zhou J, et al. A dual reinforcement learning framework for unsupervised text style transfer. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 5116–512
20 Zhang Y, Xu J, Yang P, et al. Learning sentiment memories for sentiment modification without parallel data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 1103–1108
21 Malmi E, Severyn A, Rothe S. Unsupervised text style transfer with padded masked language models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Online, 2020. 8671–8680
22 Prabhumoye S, Tsvetkov Y, Salakhutdinov R, et al. Style transfer through back-translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 866–876
23 Wang K, Hua H, Wan X. Controllable unsupervised text attribute transfer via editing entangled latent representation. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, 2019. 11036–11046
24 He J, Wang X, Neubig G, et al. A probabilistic formulation of unsupervised text style transfer. In: Proceedings of International Conference on Learning Representations, Addis Ababa, 2020
25 Liu Y, Neubig G, Wieting J. On learning text style transfer with direct rewards. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 2021. 4262–4273
26 Xiao F, Pang L, Lan Y, et al. Transductive learning for unsupervised text style transfer. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Punta Cana, 2021. 2510–2521
27 Wu C, Socher R, Xiong C. Global-to-local memory pointer networks for task-oriented dialogue. In: Proceedings of the 7th International Conference on Learning Representations, 2019
28 Yang P, Li L, Luo F, et al. Enhancing topic-to-essay generation with external commonsense knowledge. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 2002–2012
29 Ayana, Wang Z Y, Xu L, et al. Topic-sensitive neural headline generation. Sci China Inf Sci, 2020, 63: 182103
30 Yu W, Zhu C, Li Z, et al. A survey of knowledge-enhanced text generation. 2020. ArXiv:2010.04389
31 Rousseeuw P J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math, 1987, 20: 53–65
32 Rao S, Tetreault J. Dear sir or madam, may I introduce the GYAFC dataset: corpus, benchmarks and metrics for formality style transfer. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Louisiana, 2018. 129–140
33 Liu D, Fu J, Zhang Y, et al. Revision in continuous space: unsupervised text style transfer without adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, New York, 2020. 8376–8383
34 Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, 2019. 4171–4186
35 Loper E, Bird S. NLTK: the natural language toolkit. In: Proceedings of the International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, Sydney, 2002. 63–70
36 Heafield K. KenLM: faster and smaller language model queries. In: Proceedings of the 6th Workshop on Statistical Machine Translation, 2011. 187–197
37 Li J, Galley M, Brockett C, et al. A diversity-promoting objective function for neural conversation models. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016. 110–119
38 Pennington J, Socher R, Manning C D. Glove: global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1532–1543
39 Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations, San Diego, 2015
40 Guo Q P, Qiu X P, Xue X Y, et al. Syntax-guided text generation via graph neural network. Sci China Inf Sci, 2021, 64: 152102
41 Bai G R, He S Z, Liu K, et al. Example-guided stylized response generation in zero-shot setting. Sci China Inf Sci, 2022, 65: 149103
42 dos Santos C, Melnyk I, Padhi I. Fighting offensive language on social media with unsupervised text style transfer. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 189–194
43 Tran M, Zhang Y, Soleymani M. Towards a friendly online community: an unsupervised style transfer framework for profanity redaction. In: Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, 2020. 2107–2114