Self-supervised Text Style Transfer Using Cycle-Consistent Adversarial Networks

Self-supervised Text Style Transfer using Cycle-Consistent Adversarial Networks

MORENO LA QUATRA*, Kore University of Enna, Italy
GIUSEPPE GALLIPOLI*, Politecnico di Torino, Italy
LUCA CAGLIERO, Politecnico di Torino, Italy

Text Style Transfer (TST) is a relevant branch of natural language processing that aims to control the style attributes of a piece of text while preserving its original content. To address TST in the absence of parallel data, Cycle-consistent Generative Adversarial Networks (CycleGANs) have recently emerged as promising solutions. Existing CycleGAN-based TST approaches suffer from the following limitations: (1) They apply self-supervision, based on the cycle-consistency principle, in the latent space. This approach turns out to be less robust to mixed-style inputs, i.e., when the source text is partly in the original and partly in the target style; (2) Generators and discriminators rely on recurrent networks, which are exposed to known issues with long-term text dependencies; (3) The target style is weakly enforced, as the discriminator distinguishes real from fake sentences without explicitly accounting for the generated text's style. We propose a new CycleGAN-based TST approach that applies self-supervision directly at the sequence level to effectively handle mixed-style inputs and employs Transformers to leverage the attention mechanism for both text encoding and decoding. We also employ a pre-trained style classifier to guide the generation of text in the target style while maintaining the original content's meaning. The experimental results achieved on the formality and sentiment transfer tasks show that our approach outperforms existing ones, both CycleGAN-based and not (including an open-source Large Language Model), on benchmark data and shows better robustness to mixed-style inputs.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Natural language processing** → **Natural language generation**;

Additional Key Words and Phrases: Text Style Transfer, Sentiment transfer, Formality transfer, Cycle-consistent Generative Adversarial Networks, Transformers

ACM Reference Format:
Moreno La Quatra, Giuseppe Gallipoli, and Luca Cagliero. 2024. Self-supervised Text Style Transfer using Cycle-Consistent Adversarial Networks. *ACM Trans. Intell. Syst. Technol.* 1, 1 (July 2024), 37 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Language is strongly dependent on both the writers/speakers' characteristics and its context of use (e.g., time, place, scenario, intent). Although humans naturally take these factors into account, Artificial Intelligence systems could struggle to properly handle these aspects. As a result, the development of Natural Language Processing (NLP) tools that are capable of controlling the characteristics of the generated text has become particularly appealing.

*Both authors contributed equally to this research.

Authors' addresses: Moreno La Quatra, moreno.laquatra@unikore.it, Kore University of Enna, Piazza dell'Università, Enna, Italy, 94100; Giuseppe Gallipoli, giuseppe.gallipoli@polito.it, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, Italy, 10129; Luca Cagliero, luca.cagliero@polito.it, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, Italy, 10129.

2157-6904/2024/7-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

Text Style Transfer (TST) is a well-known NLP task. It focuses on changing the style attributes of a piece of text from the source style to a target one (e.g., from an informal version to its formal one) while preserving the original message conveyed by the text. Changing the text style is relevant to a wide range of real-life applications ranging from online content moderation to intelligent writing assistants [13]. TST solutions may improve the user experience by enhancing the intelligibility and pertinence of the generated text as well as adapting the language to the current situation and writer/speaker's intent [9]. Importantly, style transfer must be achieved with minimal changes to the text to preserve the original content as much as possible.

In this work, we address TST in an unsupervised scenario, i.e., we assume that there is a lack of parallel annotated data to train sequence-to-sequence models [36]. The key challenges in unsupervised TST are (1) The preservation of the original content of the source text, and (2) The correct identification and replacement of the stylistic elements present in the textual content. In the absence of parallel training data, disentangling style and content is known to be particularly challenging [16]. On the other hand, unsupervised TST approaches are, broadly speaking, more resource-efficient as they do not involve labor-intensive training tasks [47].

We propose a new architecture for TST relying on Cycle-consistent Generative Adversarial Networks (CycleGANs). CycleGANs exploit the cycle-consistency principle for self-supervised adversarial learning. In the context of TST, they have recently emerged as promising sequence-to-sequence approaches to disentangle text style and content [12].

Existing CycleGAN-based approaches to TST face the following issues [12]:

I1) **Self-supervision in the latent space**: they encode/decode the input/output text and employ fully-connected neural networks to implement the generator and discriminator models. This makes content and style information tightly connected to the text embedding representation, facing issues while coping with mixed-style content, i.e., input textual sequences that are partly in the original style (e.g., informal) and partly in the target one (e.g., formal).

I2) **Recurrent networks**: generators and discriminators consist of LSTM networks, which are known to be suboptimal for coping with long-term text dependencies. Although the interest of the NLP community has already shifted towards the use of Transformer encoder-decoder architectures [38], to the best of our knowledge existing CycleGAN-based TST approaches do not rely on Transformers yet.

I3) **Weak enforcement of the target style in adversarial learning**: since in the adversarial learning process the discriminator distinguishes between real and fake sentences without explicitly taking the style of the generated text into account, the target style is weakly enforced.

Our approach overcomes the limitations of existing approaches by introducing the following innovative features:

- **Self-supervision at the sequence level**: to overcome issue I1, it applies self-supervision, based on the cycle-consistency principle [47], directly to the raw input sequences. During the training process, the adversarial loss ensures that the generated text is indistinguishable from the target text, whereas the cycle-consistency loss ensures that the mapping between the source and target text styles is invertible.
- **CycleGANs using Transformers**: to overcome issue I2, it adopts a self-supervised approach based on CycleGANs [47] which automatically learns the mapping between the original and target styles without the need for paired data. The proposed framework consists of two generators and two discriminators. All of them are based on the Transformer architecture [38].

- **Classifier-guided text generation**: to overcome issue I3, the CycleGAN generators leverage a pre-trained classifier performing text style prediction. The classification loss returned by the classifier is integrated into the generators' loss functions to guide the text generation process. The style classifier is aimed to guide the generators to produce text with the desired style attributes while maintaining the original content's meaning.

The empirical results, achieved on benchmark TST datasets for sentiment and formality transfer, show the superior performance of the proposed approach:

- **Against state-of-the-art unsupervised TST models**: we compare the performance of our approach with that of recently proposed unsupervised approaches to TST, including Transformer-based and CycleGAN-based ones [12]. The presented architecture outperforms all the tested competitors, e.g., +6.8 points of SacreBLEU on the GYAFC dataset (see Tables 4 and 5).
- **On mixed-style inputs**: we run extensive experiments on TST tests suited to a mixed-style scenario. The results, exemplified in Figure 1, confirm the superior performance of our approach while coping with mixed-style inputs.
- **Against Large Language Models**: we also compare our approach with a state-of-the-art open-source Large Language Model with a similar number of parameters, i.e., Llama2-7B [37]. The results show that our approach averagely performs better on benchmark data and is more robust than the tested LLM to mixed-style inputs.
- **In a human evaluation**: we carried out a human evaluation to qualitatively assess the quality of a sample of TST outcomes. The results are coherent with the quantitative performance metrics.

As an example, the results summarized in Figure 1 show that CycleGAN (our approach) generates output sequences that are most syntactically similar to the expected outcome (the higher ref-BLEU the better) on all the tested configurations of mixed-style inputs. The mixing ratio X%-Y% indicates the percentage ratio of original (X%) and target (Y%) style in the input. The performance of the Large Language Model (Llama2) is closer to that of CycleGAN when there is no mix (e.g., $X \approx 0$% or $Y \approx 0$%), whereas is significantly worse in a mixed scenario (e.g., X=Y=50%).

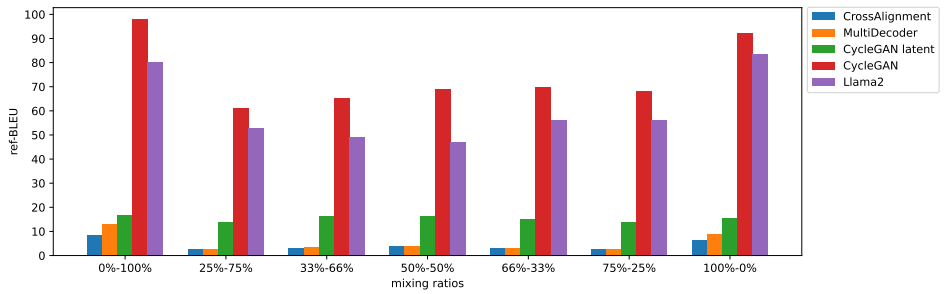

Fig. 1. Performance comparison on mixed-style inputs. Dataset: GYAFC-music. Metric: ref-BLEU. Mixing ratios are in the form X%-Y%, where X and Y are the percentages of original and target style in the input, respectively. Approaches: CycleGAN (ours), Llama2-7B [37], CycleGAN latent (variant of CycleGAN without sequence-level cycle-consistency), CrossAlignment [34], and MultiDecoder [5].

In summary, the novelty of our TST approach lies in: (1) The application of cycle-consistency directly to the input sequence, making the approach more robust for content preservation, particularly when coping with mixed-style inputs (see the results in Section 5.9); (2) The adoption

of Transformers in a CycleGAN TST approach (see the empirical comparisons in Section 5.5); and (3) The use of a style classifier to foster the generator to produce text in the target style (see Section 4.1.1 for further details).

2 RELATED WORKS

According to a recently proposed categorization [9], existing TST methods can be classified as (1) *Parallel supervised*, if they are trained on known pairs of text with different styles; (2) *Non-parallel supervised*, if the style labels are available but the matching between text pairs is missing; (3) *Purely unsupervised*, if the style labels are not available.

Parallel supervised or semi-supervised approaches (e.g., Shang et al. [33], Wang et al. [39], Xu et al. [43]) require large-scale style-to-style parallel data, i.e., examples of parallel sentences conveying the same message with different style attributes. However, their generation is extremely labor-intensive. Conversely, non-parallel supervised approaches are trained on large text corpora annotated with style labels. Relaxing the constraint of having style-to-style text pairs makes the problem challenging yet more tractable in real scenarios. This paper falls into the latter category.

Non-parallel supervised methods need to address the following issues:

- *Content preservation*: it involves maintaining the original textual content while transforming the text style. Preserving the underlying meaning, semantic information, and structural characteristics of the input text is essential to ensure the coherence and fidelity of the generated output. However, achieving effective content preservation while simultaneously changing the text style is a non-trivial task.
- *Style-content disentanglement*: it refers to the process of correctly separating the style attributes from the content in the text. This disentanglement is challenging because style and content are inherently intertwined and strongly related to each other. Modifying the style of a text without altering its content requires the model to accurately identify and manipulate the style-specific attributes while keeping the underlying content intact [16].

Style-content disentanglement can be achieved through different strategies:

- *Explicit disentanglement* [18, 42, 44]: it entails directly replacing the text with the original style attributes with new pieces of text that have the desired target style attribute. This approach explicitly separates the style and content. However, it can be applied only when style and content can easily be separated and the style transfer can be realized by changing only some selected words.
- *Implicit disentanglement* [5, 8, 26]: it learns two distinct latent representations, one for the content and the other for the style. By manipulating these separate representations, the model can ideally modify the style while preserving the content. Different techniques such as back-translation, attribute control generation and adversarial training are usually adopted to realize this approach.
- *Without disentanglement* [3, 7, 23]: the style-content separation is concealed and the model does not explicitly distinguish between them during the style transfer process. This approach aims at seamlessly transforming the style attributes while implicitly capturing and preserving the underlying content.

In our method, we adopt a strategy without disentanglement. Recent approaches to TST without disentanglement have explored the combination of linguistic graph structures and Transformer-based architectures [35]. An extensive review of existing techniques can be found in [9].

Adversarial learning has already been successfully employed to model style-content disentanglement and achieved fairly good content preservation. Recent works [1, 12, 21, 46] have already

adopted Generative Adversarial Networks (GANs) and cycle-consistency for non-parallel supervised Text Style Transfer. The key differences with the present work are summarized below.

- Zhao et al. [46] propose an encoder-decoder framework where text style and content are encoded into two distinct latent vectors (i.e., implicit disentanglement). The encoding and decoding functions are coupled with a style discrepancy loss, which models the style shift from the original domain to the target one, and with a cycle-consistency loss, which ensures content preservation. Unlike Zhao et al. [46], our approach adopts CycleGANs [47] and is without disentanglement.
- Chen et al. [1] present a GAN framework that leverages optimal transport and uses the feature mover's distance [41] as training loss. Unlike the present work, they adopt the cycle-consistency principle only for addressing the task of unsupervised deciphering in the latent feature space, relying on LSTM networks for text generation and convolutional networks as sentence feature extractors.
- Huang et al. [12] adopt CycleGANs by imposing the cycle-consistent constraint in the continuous latent space. They rely on the LSTM architecture to encode/decode the input/output sequence and employ a two-layer fully-connected neural network to implement the generator and discriminator models. In contrast, our proposed approach performs adversarial training on the raw text sequences and computes the cycle-consistency loss at the text level, allowing for more fine-grained control of the text attribute style.
- Lorandi et al. [21] focus on sentiment transfer using CycleGANs and LSTMs. In contrast, our approach explores multiple style attributes, utilizes Transformer architectures, and integrates a style classifier to enhance style transfer quality and fidelity.

Recently, some research has explored the use of Large Language Models (LLMs) to address TST. For instance, Reif et al. [30] propose an augmented zero-shot learning strategy showing promising results on various TST tasks without requiring fine-tuning or exemplars in the target style. An empirical comparison between our method and an LLM can be found in Section 5.6.

3 PRELIMINARIES

Table 1. Summary of notations for the Text Style Transfer task.

Symbol	Description
A, B	Source/Target style
$\mathcal{X}_s$	Textual corpora with style s
x_s, y_s	Input (Output) sequence in style s
$\mathcal{F}, \mathcal{G}$	Mapping function from style A (B) to B (A)
$G_{A\to B}, G_{B\to A}$	Generator from style A (B) to B (A)
D_A, D_B	Discriminator for style A (B)
SC	Style classifier
$\mathcal{L}$	Overall loss function
$\mathcal{L}_{G_{A\to B}}, \mathcal{L}_{G_{B\to A}}$	Generator loss
$\mathcal{L}_{G_{D_A}}, \mathcal{L}_{G_{D_B}}$	Adversarial loss
$\mathcal{L}_{cyc_{A\to B\to A}}, \mathcal{L}_{cyc_{B\to A\to B}}$	Cycle-consistency loss
$\mathcal{L}_{style_A}, \mathcal{L}_{style_B}$	Classifier-guided loss
$\mathcal{L}_{D_A}, \mathcal{L}_{D_B}$	Discriminator loss
$\mathcal{L}_{D_A}^{real}, \mathcal{L}_{D_A}^{fake}, \mathcal{L}_{D_B}^{real}, \mathcal{L}_{D_B}^{fake}$	Real/Fake sample discriminator loss
$\lambda_{gen}, \lambda_{cyc}, \lambda_{style}, \lambda_{dis}$	Loss scaling factors

In this section we introduce the preliminary concepts and formally state the problem under consideration. For the sake of readability, the notation used throughout the section is summarized in Table 1.

Text Style Transfer (TST). TST aims to learn a mapping function $\mathcal{F}$ that transforms an input text x_A with source style A into its transferred version x_B with target style B. Similarly, function $\mathcal{G}$ applies the reverse transformation, i.e., from x_B to x_A. Unlike style-conditioned text generation [14], in TST the transformation preserves the original content while transferring the style from A to B.

$$\mathcal{F} : x_A \rightarrow x_B \mid x_A \qquad \mathcal{G} : x_B \rightarrow x_A \mid x_B \tag{1}$$

Hereafter, we will consider the level of formality (i.e., formal or informal) or the sentiment score (i.e., positive or negative) as style attributes. The main TST complexity lies in the tight connection between content and style. For example, the level of formality of a piece of text is often determined not only by a particular linguistic register but also by other characteristics such as syntax and orthography. For the sake of simplicity, we also assume to be in a binary style transfer scenario[1].

Cycle-consistent Generative Adversarial Networks (CycleGANs). Our goal is to address TST by leveraging Cycle-consistent Generative Adversarial Networks (CycleGANs). They are a class of Generative Adversarial Networks (GANs) that can learn the mapping function between two domains without the need for parallel data [47]. Although they have been introduced in the field of Computer Vision, CycleGANs are general-purpose architectures that can be used to accomplish a variety of tasks, including TST. The use of CycleGANs enables the adoption of a self-supervised paradigm, relaxing the constraint on the availability of parallel textual data.

CycleGAN architectures typically comprise four models, including two generators and two discriminators. The generators learn the mapping functions while the discriminators ensure the quality of the generated outputs. In the following, we outline the general formulation of CycleGAN training objectives. For a detailed description of both the architecture and the training process specific to the task under consideration, please refer to Section 4.

Let X and Y be two domains with training examples $x_i \in X$ and $y_j \in Y$. The corresponding data distributions can be denoted as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. The generators F and G aim to learn the following mappings: $\mathcal{F} : X \rightarrow Y$ and $\mathcal{G} : Y \rightarrow X$. The discriminator D_Y aims to distinguish between real samples y and generated samples $F(x)$. Similarly, D_X discriminates between x and $G(y)$[2].

CycleGAN training involves two objectives: adversarial losses and cycle-consistency loss. Adversarial losses try to match the distribution of generated samples with the data distribution in the target domain. Specifically, for the mapping function $\mathcal{F} : X \rightarrow Y$, it can be expressed as follows:

$$\mathcal{L}_{\text{GAN}}(F, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(F(x)))] \tag{2}$$

It is an adversarial loss since F aims to minimize it against an adversary D_Y that tries to maximize it. Similarly, it is possible to define an adversarial loss for the mapping function $\mathcal{G} : Y \rightarrow X$.

The cycle-consistency loss constrains the mapping functions to ensure that their sequential application to the input sample x allows for its reconstruction. Additionally, it addresses the mode collapse problem [47]. By combining the reconstruction constraint of both mapping functions, it can be defined as follows:

$$\mathcal{L}_{\text{cyc}}(F, G) = \mathbb{E}_{x \sim p_{data}(x)}[\|G(F(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|F(G(y)) - y\|_1] \tag{3}$$

[1]The multiple style transfer problem is out of the scope of the present work but, as discussed in [12], can be addressed by factorizing the problem into binary subtasks.

[2]With a slight abuse of notation, $F(x)$ or $\mathcal{F}(x)$ will be used interchangeably hereafter for the sake of simplicity.

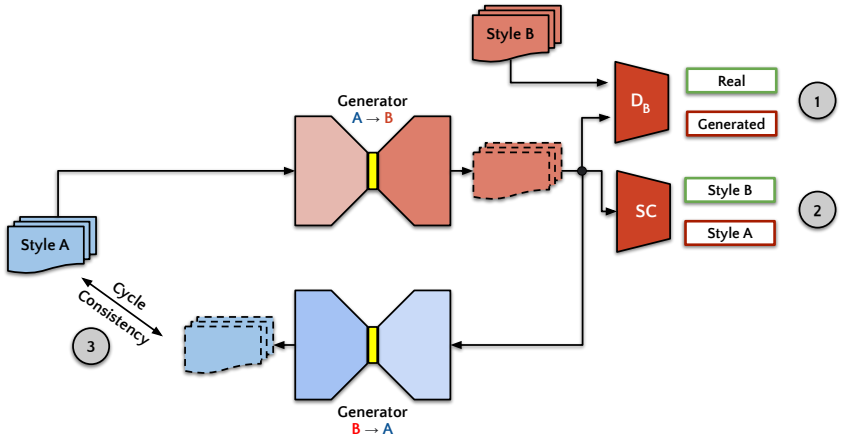

Fig. 2. The figure illustrates the training process for cycle A to B to A; the training process for cycle B to A to B is similar but the roles of the source and target texts are reversed. The generated text is reported using dashed lines and style A and style B are illustrated in blue and red, respectively. Numbered circles indicate the components of the loss function used to train the architecture.

This general formulation of CycleGAN training objectives can easily be adapted to the Text Style Transfer task. The main difference lies in defining the two domains, X and Y, as the input and target styles A and B.

4 METHOD

Figure 2 shows a sketch of the proposed method. The objective is to learn the mapping between the two styles using two generators, $G_{A\to B}$ and $G_{B\to A}$, and two discriminators, D_A and D_B. These components work together to learn the mapping between the source and target styles. A detailed description of the generator and discriminator characteristics is given below.

In addition to the generators and discriminators, we also use an external, pre-trained style classifier, hereafter denoted by SC. This model aims to classify the style of a given text sample. During the training process of the CycleGAN model, the generators receive feedback from the style classification model on the style of the generated content. This feedback is exploited by the generators to effectively produce text pieces with the desired style attribute.

4.1 Generator

The purpose of the generators $G_{A\to B}$ and $G_{B\to A}$ is to learn the transformation between the source and target pieces of texts. A modification of a specific text attribute, such as style or sentiment, must preserve the original content. The generator $G_{A\to B}$ takes a sequence of tokens, $x_A = (x_{A,1}, x_{A,2}, \ldots, x_{A,N})$, as input, where $x_{A,n}$ is the n-th token in the sequence. The output of the generator is a sequence of tokens, $y_B = (y_{B,1}, y_{B,2}, \ldots, y_{B,M})$, where $y_{B,m}$ is the m-th token in the output sequence. It is worth noting that the lengths of the input and output sequences may be different (i.e., $N \neq M$). The generator $G_{B\to A}$ handles a similar process, transforming the piece of text written in style B into a text written in style A. The key point is that the input and output sequences are not paired, and the model cannot be trained based on prior knowledge of the expected output.

The proposed method involves two cycles, $A \to B \to A$ and $B \to A \to B$, which operate as follows. For the cycle $A \to B \to A$, the generator $G_{A\to B}$ is trained to predict the output sequence y_B given the input sequence x_A. The output of this generator is then fed to the discriminator D_B,

which aims at distinguishing between samples drawn from the original distribution and those generated by the generator $G_{A\to B}$, which transfers the input text's style to the target style. The output of the generator is also fed back to the generator $G_{B\to A}$, which transforms the style of the generated text back to the original style. The output of the second generator, $y_A = G_{B\to A}(y_B)$, corresponds to the reconstructed text that should be as close as possible to the original input text.

The generators $G_{A\to B}$ and $G_{B\to A}$ are trained to minimize the following loss functions:

$$\mathcal{L}_{G_{A\to B}} = \lambda_{gen}\mathcal{L}_{G_{D_B}} + \lambda_{cyc}\,\mathcal{L}_{cyc_{A\to B\to A}} + \lambda_{style}\,\mathcal{L}_{style_B} \tag{4}$$

$$\mathcal{L}_{G_{B\to A}} = \lambda_{gen}\mathcal{L}_{G_{D_A}} + \lambda_{cyc}\,\mathcal{L}_{cyc_{B\to A\to B}} + \lambda_{style}\,\mathcal{L}_{style_A} \tag{5}$$

Here, $\mathcal{L}_{G_{D_B}}$ (illustrated in point 1 in Figure 2) and $\mathcal{L}_{G_{D_A}}$ are the adversarial losses (see Equation 2) and represent the feedback from the corresponding discriminator (i.e., the extent to which the generator is able to generate text that is indistinguishable from the target), whereas $\mathcal{L}_{cyc_{A\to B\to A}}$ (illustrated in point 3 in Figure 2) and $\mathcal{L}_{cyc_{B\to A\to B}}$ are the cycle-consistency losses (see Equation 3) computed at the end of the corresponding cycle by comparing the output of the second generator to the input sequence. More formal definitions of the discriminator and cycle losses are available in Sections 4.2 and 4.3, respectively. $\mathcal{L}_{style_B}$ and $\mathcal{L}_{style_A}$ are the style classifier losses that are computed using the pre-trained style classifier. These components of the loss function (represented in point 2 in Figure 2), aim at ensuring that the generator is able to generate text that is consistent with the target style. $\mathcal{L}_{style_B}$ and $\mathcal{L}_{style_A}$ corresponds to the binary cross-entropy loss between the predicted style and the target style (known according to the transformation being learned). The classifier-guided loss is computed using the pre-trained style classifier but only the generator is updated using this loss (e.g., the pre-trained style classifier is not trained during the adversarial training process). The classifier-guided loss can be formalized as follows:

$$\mathcal{L}_{style} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log SC(x_i) + (1-y_i)\log(1-SC(x_i))] \tag{6}$$

where N is the number of samples in the batch, x_i is the input sequence, y_i is the target label (i.e., 1 for style B and 0 for style A), and $SC(x_i)$ is the output of the style classifier (see Section 4.1.1 for further details) for the input sequence x_i classified using the `[CLS]` token. The loss is calculated by taking the average over all samples in the batch.

A different hyperparameter is associated with each of the three loss components in Equation 4 (and 5): specifically, λ_{gen}, λ_{cyc} and λ_{style} respectively control the relative importance of $\mathcal{L}_{G_{D_B}}$ ($\mathcal{L}_{G_{D_A}}$), $\mathcal{L}_{cyc_{A\to B\to A}}$ ($\mathcal{L}_{cyc_{B\to A\to B}}$) and $\mathcal{L}_{style_B}$ ($\mathcal{L}_{style_A}$).

For the cycles $A \to B \to A$ and $B \to A \to B$ the process is quite similar: the generators and discriminators operate to learn the transformation between the source and target styles.

4.1.1 Style classifier. The aim of the style classifier loss is to ensure the alignment with the target style, complementing content preservation and style transfer produced by the cycle consistency loss. It provides tailored guidance for accurate style transformation. Importantly, the discriminator, described in Section 4.2, distinguishes between real and fake sequences without taking the style of the generated output into account. Although it identifies out-of-distribution samples, the adversarial learning process weakly enforces the target style of the output text. To overcome this issue, the style classifier aims to provide explicit feedback to the generator on the style quality of the generated texts, thus mitigating the limitations of adversarial learning in TST.

4.2 Discriminator

The discriminators D_A and D_B are responsible for distinguishing between real and generated text. In line with the original GAN framework [6], the discriminators are trained to maximize the

probability of correctly classifying real and generated text. Specifically, D_A is trained to distinguish between the source texts and the output of the generator $G_{B\to A}$, where the source texts are samples drawn from the source distribution. Similarly, D_B is trained to distinguish between the target texts, which are samples drawn from the target distribution, and the output of the generator $G_{A\to B}$. The discriminators are trained to minimize the following loss functions:

$$\mathcal{L}_{D_A} = \mathcal{L}_{D_A}^{real} + \mathcal{L}_{D_A}^{fake} \tag{7}$$

$$\mathcal{L}_{D_B} = \mathcal{L}_{D_B}^{real} + \mathcal{L}_{D_B}^{fake} \tag{8}$$

where $\mathcal{L}_{D_A}^{real}$ and $\mathcal{L}_{D_B}^{real}$ denote the losses computed using data sampled from the source domain and the target domain, respectively, whereas $\mathcal{L}_{D_A}^{fake}$ and $\mathcal{L}_{D_B}^{fake}$ denote the losses computed using the output sequences of the generators $G_{B\to A}$ and $G_{A\to B}$, respectively. The weight of the discriminator losses in the overall objective function is controlled by a hyperparameter λ_{dis}.

Each term of the discriminator loss (i.e., $\mathcal{L}_{D_A}^{real}$, $\mathcal{L}_{D_A}^{fake}$, $\mathcal{L}_{D_B}^{real}$, and $\mathcal{L}_{D_B}^{fake}$) is defined as a Binary Cross-Entropy loss which, for a given discriminator D, is given by:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log D(x_i) + (1-y_i)\log(1-D(x_i))] \tag{9}$$

where N is the number of samples in the batch, x_i is the input sequence, y_i is the target label (i.e., 1 for real text and 0 for generated text), and $D(x_i)$ is the output of the discriminator for the input sequence x_i classified using the `[CLS]` token. The loss is calculated by taking the average over all samples in the batch.

The adversarial losses computed using the output of the discriminators are back-propagated to the generators, allowing them to learn to generate text that is consistent with the data sampled from the target domain. By utilizing Transformer-based models as discriminators, we can efficiently learn the text style consistency within the target domain, thus improving the overall effectiveness of the training process.

4.3 Cycle consistency

The goal of the proposed method is to learn the mapping between the source and target domains. In Figure 2 we illustrate the process for the case in which the source domain is A while B is the target domain. However, the process is analogous the other way around. Given an input sequence x_A in the source domain, the generator $G_{A\to B}$ is trained to generate a sequence y_B in the target domain. However, due to the lack of parallel annotated data in the target domain during training, the generator $G_{A\to B}$ is unable to directly learn the mapping between x_A and y_B. To address this issue, the cyclic architecture first generates a sequence y_B in the target domain and then transforms it back to the source domain using the generator $G_{B\to A}$. The output of such a generator, y_A, is then compared to the input sequence x_A using a cycle-consistency loss. This loss is computed using the cross-entropy loss between the output of the second generator and the input sequence and is used to train the generator $G_{B\to A}$.

Specifically, each generator is a sequence-to-sequence model that is trained to minimize the cross-entropy loss between the generated sequence and the target sequence defined as follows:

$$\mathcal{L}_{cyc} = -\frac{1}{N \cdot T_{\text{total}}}\sum_{n=1}^{N}\sum_{t=1}^{T-1} y_{nt}\log(p_{t|t-1}) \tag{10}$$

where N is the number of samples in the batch, T_{total} is the total number of tokens across all samples in the batch, T is the length of the sequence, y_{nt} is the target token at position t in the sequence n,

and $p_{t|t-1}$ is the probability of the token at position t given the previous tokens in the sequence. The loss is calculated by taking the average over all samples in the batch.

Given the self-supervised nature of our method, the target sequence is not available during the initial transformation from the source domain to the target domain (i.e., $A \rightarrow B$). The subsequent transformation from the target domain back to the source domain (i.e., $B \rightarrow A$) aims to reconstruct the original input sequence. At this stage, the expected output is the input sequence x_A, which can be used to compute the cycle-consistency loss. Therefore, the cycle-consistency loss is computed using both the output of the generator $G_{B \rightarrow A}(y_B) = y_A$ and the input sequence x_A (for the cycle $A \rightarrow B \rightarrow A$). A similar process occurs for the cycle $B \rightarrow A \rightarrow B$.

4.4 Objective function

The full objective function is a combination of various loss functions, with each component contributing to a specific aspect of the Text Style Transfer task. The loss functions include the generator loss, the cycle-consistency loss, the style loss, and the discriminator loss, each weighted by a hyperparameter λ. The final formulation can be expressed as follows:

$$\begin{aligned} \mathcal{L}\left(G_{A \rightarrow B}, G_{B \rightarrow A}, D_A, D_B\right) = & \lambda_{gen} \mathcal{L}_{G_{D_B}} + \lambda_{cyc}\, \mathcal{L}_{cyc_{A \rightarrow B \rightarrow A}} + \lambda_{style}\, \mathcal{L}_{style_B} \\ & + \lambda_{gen} \mathcal{L}_{G_{D_A}} + \lambda_{cyc}\, \mathcal{L}_{cyc_{B \rightarrow A \rightarrow B}} + \lambda_{style}\, \mathcal{L}_{style_A} \\ & + \lambda_{dis} \mathcal{L}_{D_A} + \lambda_{dis} \mathcal{L}_{D_B} \end{aligned} \tag{11}$$

It includes the adversarial losses for both generators ($G_{A \rightarrow B}$ and $G_{B \rightarrow A}$), and discriminators (D_A and D_B), the cycle-consistency losses for both style transfer directions, and the style losses for both domains. Additionally, weighting factors (λ_{gen}, λ_{dis}, λ_{cyc} and λ_{style}) are used to balance the importance of each component in the overall objective.

It is worth noting that, while it is possible to have separate weighting factors for each direction, we employ identical weighting hyperparameters for both style transfer directions to maintain simplicity and minimize the complexity of configuration options. This choice allows us to avoid the need for justifying or making any prior assumption concerning distinct values for each direction. By ensuring uniformity in the weighting factors, we establish a balanced optimization process that treats both directions equally. The proposed implementation can easily be extended to accommodate different weighting factors for each style transfer direction if required by the specific use case. Finally, we formulate the overall optimization problem as follows:

$$G^*_{A \rightarrow B}, G^*_{B \rightarrow A} = \arg \min_{G_{A \rightarrow B}, G_{B \rightarrow A}} \max_{D_A, D_B} \mathcal{L}\left(G_{A \rightarrow B}, G_{B \rightarrow A}, D_A, D_B\right) \tag{12}$$

which expresses the min-max game played between each pair of generator-discriminator models [47].

4.5 Extension to Multiple Styles

The current approach is designed to handle only a specific pair of source and target styles. A straightforward method to handle more than two styles is to train separate pairwise TST architectures. However, this leads to scalability issues. An alternative, more efficient solution is to prompt the generator with specific instructions [2] indicating the desired style transformation. For example, by using purposefully crafted prompt tokens like `[A->B]` for converting text from style A to style B, or `[A->C]` for converting text from style A to style C, each generator can be trained to handle multiple style conversions. This approach maintains the self-supervised nature of the architecture, enabling generators to convert from any style to any other style. However, implementing this method requires careful consideration of model training and architecture adjustments, which are beyond the scope of the current work (see Section 6 for a discussion of the future research lines).

5 EXPERIMENTAL EVALUATION

We evaluate the performance of the proposed method and compare it against recent TST approaches on benchmark data. We also perform various ablation studies to evaluate the following aspects: cycle-consistency in the latent space, impact of the cycle-consistency loss coefficient, and effect of the pre-trained style classifier.

To foster the reproducibility of our results, the models and code used for the implementation of the proposed framework are publicly available, for research purposes only, at https://github.com/gallipoligiuseppe/TST-CycleGAN under the license CC BY-NC-SA.

5.1 Datasets

We consider three benchmark datasets related to two different TST tasks, i.e., sentiment transfer and formality transfer.

Sentiment transfer. The Yelp dataset [34] collects restaurant reviews. Based on their rating, reviews are labeled as positive or negative (a rating of 4 or 5 corresponds to a positive label, whereas a rating below 3 is negative). The dataset includes a test set with four human references per sentence. Train and validation sets are suited to non-parallel supervised TST as they are annotated with style attributes but the matching between text pairs is missing. For the sake of reproducibility, we use the same train/validation/test splits as in Li et al. [18] (see Table 2).

Table 2. Yelp dataset statistics.

	# train	# validation	# test
negative	177,218	2,000	500
positive	266,041	2,000	500
total	443,259	4,000	1,000

Formality transfer. Grammarly's Yahoo Answers Formality Corpus (GYAFC) [28] is a parallel dataset consisting of informal-to-formal sentence pairs. It comprises sentences from two different domains, i.e., Family & Relationships (family, in short) and Music & Entertainment (music, in short). Although the dataset includes parallel sentences, to simulate the scenario of self-supervised style transfer we select only the source sentences from the train set. The validation and test sets, on the other hand, include annotated sentences for both domains and are used to evaluate the performance of the proposed method (see Table 3).

Table 3. GYAFC dataset statistics.

		# train	# validation	# test
family	informal	51,967	2,788	1,332
	formal	51,967	2,247	1,019
	total	103,934	5,035	2,351
music	informal	52,595	2,877	1,416
	formal	52,595	2,356	1,082
	total	105,190	5,233	2,498

5.2 Metrics

We evaluate our model using a suite of established evaluation metrics [7, 23, 31]. Specifically, to quantify content preservation we compute the SacreBLEU score [25] between the system outputs and the four human references[3].

To evaluate the effectiveness of our approach for Text Style Transfer, we fine-tune a BERT-base binary classifier [4] to compute the style accuracy metric. To distinguish it from the style classifier used during model training, hereafter we will denote it by the *oracle classifier*. Its purpose is to accurately classify the style of the input text and provide a reliable evaluation metric for the quality of the generated text's style transfer. On the analyzed datasets, the oracle classifier respectively achieves the accuracies of 98.5% (Yelp), 94.0% (GYAFC-family), and 94.6% (GYAFC-music). To compute the style accuracy, according to prior works we also train a TextCNN [15] as oracle classifier (beyond the BERT-base classifier). Its classification performance is, in general, satisfactory on all the tested datasets (96.5% on Yelp, 93.2% on GYAFC-family, 93.8% on GYAFC-music) and comparable to that of the BERT-base model. Both BERT-base and TextCNN classifiers were trained on the same TST datasets under analysis: they were trained and tested on the corresponding training and test splits, respectively.

Finally, we also provide a comprehensive performance score by computing the geometric mean (GM) and harmonic mean (HM) of the SacreBLEU and style accuracy scores.

5.3 Configuration settings

We implemented the proposed architecture using the Hugging Face Transformers library [40]. As reference models for the generators and discriminators, we consider BART-base [17] (140M parameters) and DistilBERT [32] (66M parameters), respectively. For the Yelp dataset we use the case-insensitive variant of DistilBERT, whereas for GYAFC we use the case-sensitive version as the input data contains case-sensitive text. We also run experiments with larger generator models prioritizing the use of more powerful models for the most challenging (and resource-demanding) text generation task. Specifically, for the generators we also consider BART-large (400M parameters) and T5 [27] (with 60M, 220M, and 770M parameters for the small, base, and large versions, respectively). As style classifier, we use a fine-tuned BERT-base (110M parameters) model. Note that although we use the same model as for the *oracle classifier*, this is not necessarily the case. The proposed TST architecture can be trained end-to-end, enabling the simultaneous learning of the style transfer functions in both directions.

We use the validation SacreBLEU score as the reference metric to identify the best-performing training configurations and the optimal model checkpoints. Then, we evaluate them on the test set. The SacreBLEU score is calculated separately for each TST direction and the average of these two values is used as an overall score. Note that for the Yelp dataset human references are not available for the validation set. To overcome this issue, we optimize the geometric mean of the SacreBLEU score calculated between the system outputs and the source sentences, and the style accuracy.

Training details. Similar to [12], we train the model in a self-supervised setting for both tasks, even though the GYAFC dataset is a parallel corpus. Thus, for our purposes, the alignments between sentence pairs are neglected.

Based on our preliminary experiments, we observed that the impact of the hyperparameters λ_{gen}, λ_{dis} and λ_{style} is negligible. Therefore, for the sake of simplicity, hereafter we will set $\lambda_{gen} = \lambda_{dis} = \lambda_{style} = 1$. We tune the following two hyperparameters: the learning rate and the loss scaling factor λ_{cyc}. The learning rate, which controls the magnitude of the weight updates during training, is

[3]We adopt the implementation of the `sacrebleu` metric available at https://github.com/mjpost/sacreBLEU.

updated using a linear scheduler, which linearly decreases the learning rate from a maximum value, as reported in Appendix, to zero during the training process. Meanwhile, the λ_{cyc} hyperparameter controls the weight of the cycle-consistency loss in the overall objective function.

Given the computational demands of training such models and to reduce the number of configurations to be tested, we explore the hyperparameter space by considering values in the range $\left[10^{-5}, 10^{-3}\right]$ for the learning rate and $\{0.1, 1, 10\}$ for the loss scaling factor λ_{cyc}. The optimal hyperparameter values used throughout the experiments are reported in Appendix. It is worth noticing that the selection of appropriate hyperparameters may affect the performance of the model. Moreover, these hyperparameters were found to be optimal for the specific datasets and models used in our experiments, and they may not necessarily generalize to other datasets or models. The optimal values were determined through a combination of manual tuning and grid search, by evaluating the model's performance of various hyperparameter combinations on the validation sets.

To balance computational efficiency and model performance, we set the maximum input sequence length to 64 since the average number of tokens is 8.88±3.64 and 10.68±4.12 for the Yelp and GYAFC datasets, respectively, and the batch size to 128 for BART-base, 32 for BART-large, 128 for T5-small, 64 for T5-base, and 8 for T5-large. We employ the AdamW optimizer [22] with β_1 0.9, β_2 0.999, and weight decay 10^{-2}.

To ensure consistent experimental conditions and hardware utilization, we utilize a single NVIDIA® V100 GPU with 32 GB of VRAM for both training and inference of all models.

5.4 Baselines

Existing unsupervised TST methods. We test the following methods: RetrieveOnly, DeleteOnly, DeleteAndRetrieve, and TemplateBased [18], BackTranslation [26], StyleEmbedding and MultiDecoder [5], CrossAlignment [34], UnpairedTranslation [42], UnsupervisedMT [45], DualRL [23], NASTLatentLearn [11], DeepLatent [7], ReinfRewards [31], MixAndMatch [24], MultiClass [3], FineGrainedST [19], LevenshteinEdit [29], GTAE [35], CycleAutoEncoder [12], and TextGANPG [21][4]. Notice that we disregard existing supervised approaches to formality transfer because we deem their comparison with unsupervised methods unfair.

Cycle-consistency in the latent space. A key property of our approach is that it performs style transfer directly at the sequence level. Conversely, previous CycleGAN-based TST approaches apply transformations on the latent space. To evaluate our method's effectiveness against this approach, we explore two variants that conduct style transfer in the latent space. In these variants, we leverage the embedding space for style transfer. We decompose the generator network into encoder E and decoder D components, introducing two baseline models:

(1) **Sentence-level**: this approach focuses on aligning representations generated by the encoders E_A and E_B with each other. Considering the case $A \rightarrow B \rightarrow A$, this is achieved by minimizing the L1 loss between the embeddings of the input sequence encoded by E_A and its corresponding version, predicted in the target style, and then encoded by E_B. To obtain sentence representations from token representations we use average pooling. The rationale behind this approach is to ensure that the semantic meaning of the input sequence is preserved while transferring its style.

(2) **Token-level**: in this baseline model, the focus is on preserving the content of the text at the token level by maximizing the similarity between the original input and the reconstructed

[4]To ensure a fair comparison, we recompute the results of the baseline methods using our own evaluation scripts. When the baseline methods produce lowercase outputs, we lowercase the human references and retrain the style classifiers on the lowercase versions of the datasets. Lower-cased *oracle classifiers* accuracies: 90.0% (GYAFC-family) and 91.1% (GYAFC-music); TextCNN models: 89.2% (GYAFC-family) and 88.8% (GYAFC-music).

Table 4. Results on the GYAFC-family dataset – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
RetrieveOnly [18]	4.4	**92.9**	20.2	8.4	47.7	14.5	8.1
BackTranslation [26]	6.1	44.4	16.5	10.7	48.1	17.1	10.8
StyleEmbedding [5]	12.1	30.6	19.2	17.3	43.1	22.8	18.9
MultiDecoder [5]	16.1	25.6	20.3	19.8	42.8	26.3	23.4
CrossAlignment [34]	8.1	61.6	22.3	14.3	46.2	19.3	13.8
UnpairedTranslation [42]	5.2	58.1	17.4	9.5	47.2	15.7	9.4
DeleteOnly [18]	27.9	28.0	27.9	27.9	47.2	36.3	35.1
DeleteAndRetrieve [18]	21.0	52.4	33.2	30.0	45.4	30.9	28.7
TemplateBased [18]	31.9	39.4	35.4	35.2	46.0	38.3	37.7
UnsupervisedMT [45]	30.6	65.1	44.6	41.6	45.3	37.2	36.5
DualRL [23]	36.6	53.2	44.1	43.3	41.7	39.1	38.9
NASTLatentLearn [11]	38.6	49.3	43.6	43.3	43.1	40.8	40.7
CycleGAN BART (base)	43.7	50.7	47.1	46.9	49.4	46.5	46.4
CycleGAN BART (large)	43.5	50.8	47.0	46.9	49.9	46.6	46.5
CycleGAN T5 (small)	42.1	38.7	40.4	40.3	39.6	40.8	40.8
CycleGAN T5 (base)	44.0	47.7	45.8	45.8	46.2	45.1	45.1
CycleGAN T5 (large)	**45.4**	59.5	**52.0**	**51.5**	**58.1**	**51.4**	**51.0**

output. To achieve this, it minimizes the L1 loss between the token embeddings of the input sequence and its reconstructions. The token-level embeddings for the input sequence are obtained immediately after tokenization, before being fed into the model. The reconstructed tokens are taken from the output of the decoder after completing the cycle (i.e., $A \rightarrow B \rightarrow A$) in the CycleGAN architecture.

To assess the performance of the two described latent-based variants of our approach, we present experimental results in Section 5.7.

5.5 Evaluation and Comparison

Here we summarize the main results of the empirical evaluations and performance comparisons separately for each style transfer domain. We also conduct a qualitative analysis of the generated outputs whose results are provided in Appendix.

Formality transfer. Tables 4 and 5 report the performance of our method variants (denoted by the prefix name *CycleGAN*) and the baselines on the family and music domains of the GYAFC dataset, respectively. The music domain has been shown to be more challenging and, in general, less explored by previous TST studies than the family one. In both domains, our approach based on *T5 large* performs best in terms of SacreBLEU scores compared to all the tested prior works. More specifically, CycleGAN outperforms the other approaches in terms of ref-BLEU score (+6.8 vs. the best-performing competitor), showing a higher capability of content preservation and a better fluency of the generated text. Conversely, models achieving the highest accuracy scores significantly perturb the original content as the corresponding ref-BLEU scores are fairly low, resulting in a less faithful reproduction of the original meaning. Instead, the proposed approach achieves the best balance between content preservation and style transfer. Among the tested CycleGAN variants, those relying on larger generator models produce, as expected, consistently better ref-BLEU results than the other ones.

Table 5. Results on the GYAFC-music dataset – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy. * denotes results from the paper.

	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
DeepLatent [7]	21.4	**77.8**	40.8	33.6	56.2	34.7	31.0
ReinfRewards* [31]	29.2	–	–	–	–	–	–
MixAndMatch* [24]	27.7	–	–	–	–	–	–
CycleGAN BART (base)	43.6	57.2	49.9	49.5	57.8	50.2	49.7
CycleGAN BART (large)	42.0	43.1	42.5	42.5	43.8	42.9	42.9
CycleGAN T5 (small)	40.6	37.9	39.2	39.2	39.0	39.8	39.8
CycleGAN T5 (base)	42.0	45.4	43.7	43.6	47.7	44.8	44.7
CycleGAN T5 (large)	**45.6**	70.5	**56.7**	**55.4**	**70.1**	**56.5**	**55.3**

More detailed results on the most common formality transfer case, i.e., from informal to formal style, are given in Appendix. The results confirm the superior performance of CycleGAN T5 large compared to all the other methods (e.g., ref-BLEU +34.1 against ReinfRewards on GYAFC-music).

Sentiment transfer. Table 6 reports the results of our method variants (denoted by the prefix name *CycleGAN*) and the baselines on the Yelp dataset. Our method shows performance superior to all the other methods in terms of average SacreBLEU metric using the BART large model (CycleGAN 56.5 vs. 54.9 of the best-performing competitor). The better ability to preserve the original content is partly mitigated by the lower style accuracy which is, however, less critical for the sentiment transfer task (e.g., CycleGAN ≈75% accuracy in sentiment transfer vs. ≈50% in formality transfer). In fact, sentiment transfer commonly requires minimal modifications of the text to change its polarity.

In general, we claim that our model is able to achieve better content preservation thanks to the cycle-consistent structure of our architecture which is instrumental in preventing inappropriate or unnecessary modifications of the input text.

5.6 Formality and Sentiment transfer with Large Language Models

We perform an empirical comparison between our approach and a state-of-the-art open-source Large Language Model, i.e., Llama2 model [37]. Specifically, we consider the 7B version to ensure a fair comparison in terms of model size with the proposed architecture[5]. We employ it in both zero-shot and few-shot settings: in the latter case, we experiment with varying number of examples $k \in \{1, 3, 5, 10\}$ provided as input to the model. Few-shot examples consist of both the input sentence and the corresponding expected output in the target style. Examples are randomly selected from the parallel training sets for formality transfer datasets, whereas in the case of sentiment transfer, since no parallel data is available, we manually annotate the expected outputs for the selected examples.

Based on preliminary experiments, we set the model's temperature hyperparameter at 0.6. We provide the LLM with the following prompt:

```
Transform the following sentence from [SRC] style to [TGT] style.
Apply only minimal changes and preserve the meaning of the sentence.
Here you can find some examples of sentences in [SRC] style and corresponding
sentences in [TGT] style:
```

[5]Due to computational constraints, we employ a 16-bit quantization.

Table 6. Results on the Yelp dataset – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy. * denotes results from the paper.

	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
RetrieveOnly [18]	6.4	**99.9**	25.3	12.0	93.0	24.4	12.0
BackTranslation [26]	10.5	94.6	31.5	18.9	95.3	31.6	18.9
StyleEmbedding [5]	42.2	7.9	18.3	13.3	8.8	19.3	14.6
MultiDecoder [5]	29.1	46.8	36.9	35.9	50.1	38.2	36.8
CrossAlignment [34]	20.0	72.8	38.2	31.4	74.9	38.7	31.6
UnpairedTranslation [42]	33.1	50.4	40.8	39.9	51.4	41.2	40.3
DeleteOnly [18]	29.1	85.3	49.8	43.4	87.9	50.6	43.7
DeleteAndRetrieve [18]	30.0	89.9	51.9	44.9	90.6	52.1	45.1
TemplateBased [18]	39.7	83.6	57.6	53.8	85.3	58.2	54.2
UnsupervisedMT [45]	41.3	95.7	62.9	57.7	**97.4**	63.4	58.0
MultiClass [3]	51.8	86.1	66.8	64.7	87.2	67.2	65.0
DualRL [23]	51.5	88.5	**67.5**	**65.1**	90.1	**68.1**	65.5
DeepLatent [7]	40.4	83.8	58.2	54.5	86.2	59.0	55.0
FineGrainedST [19]	16.2	91.4	38.5	27.5	91.5	38.5	27.5
LevenshteinEdit [29]	48.9	87.2	65.3	62.7	82.9	63.6	61.5
GTAE [35]	51.1	86.7	66.5	64.3	85.9	66.2	64.1
NASTLatentLearn [11]	54.9	78.4	65.6	64.6	81.8	67.0	**65.7**
MixAndMatch [24]	46.6	88.3	64.1	61.0	81.7	61.7	59.3
CycleAutoEncoder* [12]	22.5	–	–	–	86.9	44.2	35.7
TextGANPG* [21]	32.4	68.0	46.9	43.9	–	–	–
CycleGAN BART (base)	55.7	78.8	66.3	65.3	77.8	65.8	64.9
CycleGAN BART (large)	**56.5**	75.1	65.1	64.5	74.6	64.9	64.3
CycleGAN T5 (small)	53.0	78.0	64.3	63.1	78.2	64.4	63.2
CycleGAN T5 (base)	54.2	76.6	64.4	63.5	77.3	64.7	63.7
CycleGAN T5 (large)	55.3	72.9	63.5	62.9	73.7	63.8	63.2

```
Input ([SRC] style): [SRC_EXi]
Output ([TGT] style): [TGT_EXi]
...
Input ([SRC] style): [SRC_INPUT]
Output ([TGT] style):
```

where we replace `[SRC]` and `[TGT]` with the actual source and target styles, `[SRC_EXi]` and `[TGT_EXi]` with the source and target sentences for each of the k examples (for $k > 0$), and `[SRC_INPUT]` with the current test sample to be transferred.

Table 7 reports the results achieved for both formality and sentiment transfer tasks, while more detailed results for the informal-to-formal transfer can be found in Appendix. In both tasks, the ref-BLEU and accuracy performance generally increases while providing more input examples until reaching a steady state. This is probably due to the fact that, when providing numerous examples, some noise may be introduced, potentially misleading the model. Surprisingly, the ref-BLEU results on the Yelp dataset for $k = 1, 3$ are worse than in the zero-shot setting. One possible explanation is that, since in the sentiment transfer task style can often be transferred by modifying only a few words, in the zero-shot setting the model may tend to apply fewer modifications, resulting in a higher ref-BLEU score but lower accuracy. In the 1- or 3-shot settings, the accuracy increases at

Table 7. Results on the GYAFC-family, GYAFC-music, and Yelp datasets of Llama2-7B-Chat model for varying number of examples k in the 0/few-shot setting – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy. ★ denotes the results of our best models.

dataset	k	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
GYAFC-family	0	24.0	80.4	43.9	36.9	77.0	43.0	36.6
	1	21.9	85.2	43.2	34.9	82.8	42.5	34.6
	3	36.8	87.5	**56.7**	**51.8**	86.9	**56.5**	**51.7**
	5	37.2	84.6	56.1	51.7	83.5	55.7	51.4
	10	34.0	**88.3**	54.8	49.1	**87.5**	54.5	49.0
	★	**45.4**	59.5	52.0	51.5	58.1	51.4	51.0
GYAFC-music	0	26.0	78.1	45.1	39.0	73.3	43.6	38.4
	1	34.0	**95.0**	56.8	50.1	91.8	55.8	49.6
	3	40.1	94.1	**61.5**	**56.3**	**91.9**	**60.7**	**55.8**
	5	37.4	92.2	58.7	53.2	90.4	58.1	52.9
	10	38.3	92.1	59.4	54.1	90.2	58.8	53.7
	★	**45.6**	70.5	56.7	55.4	70.1	56.5	55.3
Yelp	0	42.7	83.2	59.6	56.5	79.6	58.3	55.6
	1	34.0	92.4	56.1	49.7	88.9	55.0	49.2
	3	36.1	**92.9**	57.9	52.0	**89.4**	56.8	51.4
	5	43.1	91.2	62.7	58.5	87.4	61.4	57.7
	10	53.4	84.8	**67.3**	**65.5**	82.9	**66.5**	**64.9**
	★	**56.5**	75.1	65.1	64.5	74.6	64.9	64.3

the expense of a lower ref-BLEU score. This is likely because the model requires more examples to adhere to the requirement of applying only minimal changes to the input sentences.

By comparing LLM results with those of our best models, we can state that our method consistently outperforms Llama2 in terms of content preservation on both tasks (i.e., +8.2 on GYAFC-family, +5.5 on GYAFC-music, +3.1 on Yelp). In contrast, Llama2 achieves the highest accuracy scores, even when compared to the other baselines in the formality transfer task. The results highlight that the TST performance of Llama2 is fair without ad hoc fine-tuning. It is also worth noting that model fine-tuning would require parallel data and thus is out of the scope of the present work. In conclusion, our proposed approach confirms its superior performance in content preservation, even when compared to a larger and more powerful Large Language Model.

5.7 Cycle-consistency in the Latent Space: Sentence-Level vs. Token-Level

In this section, we present the results of an ablation study conducted to compare the performance of the latent-based versions of our approach (described in Section 5.4). The purpose is to quantitatively compare these model variants with the proposed solution, which enforces the cycle-consistency constraint directly to the raw input sequence. To better isolate the effect of the cycle-consistency loss, we exclude the pre-trained style classifier and its corresponding loss term. Additionally, to ensure a fair comparison, we use the same generator model that achieved the best results in the corresponding task (i.e., T5 large for formality transfer and BART large for sentiment transfer).

The results on both tasks are shown in Table 8. As can be seen, the non-latent version (i.e., applying cycle-consistency on the raw input sequence) significantly outperforms both latent-based versions in terms of geometric mean and harmonic mean on both tasks. Upon closer inspection, in the formality transfer task, our approach achieves the best ref-BLEU scores, exhibiting an

Table 8. Ablation study. Results on the GYAFC-family, GYAFC-music, and Yelp datasets of latent-based cycle-consistency losses – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

dataset	model	latent	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
GYAFC-family	CycleGAN T5 (large)	–	**44.4**	**49.2**	**46.7**	**46.7**	**47.9**	**46.1**	**46.1**
		sentence	8.0	28.9	15.2	12.5	33.7	16.4	12.9
		token	5.9	43.3	16.0	10.4	43.5	16.0	10.4
GYAFC-music	CycleGAN T5 (large)	–	**43.3**	43.0	**43.1**	**43.1**	45.1	**44.2**	**44.2**
		sentence	8.1	23.1	13.7	12.0	36.3	17.2	13.2
		token	6.3	**59.1**	19.3	11.4	**59.1**	19.3	11.4
Yelp	CycleGAN BART (large)	–	56.9	73.9	**64.8**	**64.3**	73.1	**64.5**	**64.0**
		sentence	**58.6**	1.6	9.7	3.1	3.6	14.5	6.8
		token	0.7	**99.7**	8.4	1.4	**98.6**	8.3	1.4

improvement of more than +35.0 points on both domains. Considering style accuracy, the latent token-level version performs the best on the GYAFC-music dataset. However, it must be noted that the corresponding ref-BLEU score is extremely low. In the sentiment transfer task, the sentence-level and token-level versions achieve the best results in ref-BLEU and style accuracy, respectively. Nonetheless, they show remarkably low results in the other metric of interest (i.e., sentence-level accuracy=1.6, token-level ref-BLEU=0.7).

After manually inspecting the generated outputs, we observed that in the sentence-level version, in most cases, the input is simply copied to the output. The loss function used to train the model aims at minimizing the discrepancy between the embedding of the input and transferred sentence, therefore preserving the meaning. However, in the formality transfer task, where the output often contains multiple copies of the input, there is a low overlap with the target sentence. In contrast, in the sentiment transfer task, the input is copied to output without repetitions. Given that sentiment transfer typically involves modifying only a few words, the sentence-level version achieves a high ref-BLEU score. Considering accuracy scores, since the input text is not modified in the sentence-level version, its performance is low, especially in the sentiment transfer task (i.e., accuracy=1.6). For the token-level version, the outputs are degenerate, i.e. the model almost generates the same sentence. Consequently, the ref-BLEU scores are particularly low (e.g., 0.7 on the Yelp dataset), while the accuracy is often very high (e.g., 99.7 on the Yelp dataset) if the (degenerate) sentences are classified as belonging to the target style.

Overall, these results demonstrate the significant advantage achieved by directly enforcing cycle-consistency constraints to the raw sequence, highlighting one of the main contributions of our work.

5.8 Human Evaluation

Similar to [12, 23], we conducted a human evaluation to get a qualitative feedback on the TST results. We recruited 12 volunteers, each of them meets the following criteria: she/he holds an MSc or PhD degree, is proficient in English, and has a sufficient background in the Text Style Transfer task. We randomly picked 50 test samples per dataset and style transfer direction (300 samples overall). For each source sample and target style, annotators were asked to evaluate the quality of outputs generated by different systems. The outputs were presented in random order and without disclosing the model each output was generated from. Specifically, for each task and dataset, annotators evaluated the outputs produced by the following models: CycleGAN (ours), CycleGAN latent (i.e., the latent sentence-based version of CycleGAN), Llama2, and the two corresponding best baselines.

Table 9. Human Evaluation results on the GYAFC-family, GYAFC-music, and Yelp datasets – style accuracy (Style), content preservation (Content), fluency (Fluency), average ratings (Avg) and success rate (Success). * denotes scores for which $p < 0.05$.

dataset	model	Style	Content	Fluency	Avg	Success
GYAFC-family	DualRL [23]	2.2*	3.0*	2.6*	2.6*	8.5%*
	NASTLatentLearn [11]	2.3*	2.9*	2.7*	2.6*	5.5%*
	CycleGAN latent T5 (large)	1.5*	2.1*	1.2*	1.6*	0.5%*
	Llama2-7B-Chat	**4.2***	4.6*	**4.7**	**4.5***	**73.0%***
	CycleGAN T5 (large)	3.5	**4.9**	**4.7**	4.4	55.0%
GYAFC-music	DeepLatent [7]	2.6*	1.9*	2.5*	2.3*	35.0%*
	CycleGAN latent T5 (large)	1.7*	2.6*	1.4*	1.9*	0.5%*
	Llama2-7B-Chat	**4.3***	4.0*	4.3	4.2*	63.0%*
	CycleGAN T5 (large)	4.1	**4.8**	**4.5**	**4.5**	**74.4%**
Yelp	DualRL [23]	3.1*	3.2*	3.5*	3.3*	26.0%*
	NASTLatentLearn [11]	2.9*	3.2*	3.1*	3.1*	15.0%*
	CycleGAN latent BART (large)	1.1*	3.3*	4.2*	2.9*	0.5%*
	Llama2-7B-Chat	4.1	4.0*	4.4*	4.2*	64.5%*
	CycleGAN BART (large)	**4.3**	**4.4**	**4.5**	**4.4**	**80.0%**

The output sentences were evaluated using a 5-point Likert scale based on three criteria: (1) Style accuracy, measuring the extent to which the generated sentence aligns with the target style; (2) Content preservation, assessing how effectively the content of the input sentence is preserved; and (3) Fluency, considering the overall fluency and linguistic correctness of the output text. Similar to [23] and [18], we also calculate the average across the three criteria and denote a generated output as "successful" if it receives a rating of 4 or 5 on all three criteria.

Table 9 reports the results achieved for both tasks, including a t-test for statistical significance. In the formality transfer task, our approach excels in content preservation and fluency, achieving the best performance. Moreover, it yields the highest average score and success rate on the GYAFC-music dataset. Conversely, Llama2 demonstrates the highest style transfer score for both domains, and excels in terms of average score and success rate on the GYAFC-family dataset. Notably, our approach and Llama2 outperform other systems across all metrics by a substantial margin (e.g., +2.9 and +2.0 on content preservation and style accuracy, respectively), especially the latent sentence-based version of our approach which exhibits the lowest performance. In the sentiment transfer task, our model outperforms all baselines, including Llama2, on all metrics, thus confirming the superior quality of the generated outputs. Broadly speaking, the human evaluations are mostly aligned with the quantitative results and confirm the superior performance of our approach, particularly on content preservation. Notably, we achieved exceptionally high scores in the formality transfer task (i.e., 4.9 and 4.8 on the family and music domains, respectively), highlighting its superior capability in preserving the input content, which is known to be the most challenging constraint in Text Style Transfer. In compliance with [13], we also report the Krippendorff's alpha inter-rater agreement coefficient, which equals $\alpha = 0.76$. This high score indicates a significant level of agreement among raters, reinforcing the consistency of the conducted human evaluation.

5.9 Results on Mixed-Style Inputs

We evaluated the models' ability to preserve content while transferring style on datasets with inputs composed of mixed-style text segments. The mixed-style text versions are generated by

proportionally appending pieces of text of different styles. Table 10 shows the results on the GYAFC-family dataset with a mix of formal/informal text segments with varying mixing ratios. Hereafter we will focus on formality transfer because it is more likely to have mixed-style text than in sentiment transfer cases. Additional results are available in Appendix.

Overall, the proposed model achieves the best balance of style accuracy and content preservation across different mixing ratios. On GYAFC-family it obtains the highest geometric and harmonic mean in 5 out of 6 configurations. Notably, the performance is relatively strong even in more mixed settings such as 50% − 50%, demonstrating its ability to effectively disentangle and transfer style at the segment level rather than just averaging effects across the full input.

The baseline models, namely CrossAlignment and MultiDecoder, exhibited consistently lower performance, leading to a notable decrease in overall effectiveness in mixed settings. The latent space variant of CycleGAN also lagged behind our approach, highlighting the benefit of applying the cycle-consistency constraint directly to the raw input sequences. Llama consistently came second to our proposed approach. However, its performance degraded more significantly than CycleGAN as the mixture became more balanced. This suggests CycleGAN may have an advantage in more ambiguous scenarios where the overall style is unclear.

These results demonstrate that the proposed methodology is highly effective at style transfer even when the input text contains mixtures of different styles, outperforming prior work especially on more balanced mixtures. This underscores its ability to perform style transfer at a fine-grained segment level.

5.10 Ablation studies

In this section, we delve into the results of two complementary ablation studies. The first experiment explores the impact of the λ_{cyc} scaling factor in the cycle-consistency loss, whereas the second one analyzes the effect of the pre-trained style classifier, considering both the additional loss term and the model used.

Cycle-consistency loss. In this ablation study, we investigate the impact on performance when varying the cycle-consistency loss coefficient λ_{cyc}. To better analyze and isolate the effect of this loss component, we conduct this analysis by excluding the pre-trained style classifier and its corresponding loss term. Consequently, we set $\lambda_{gen} = \lambda_{dis} = 1$, $\lambda_{style} = 0$ and experiment with different values of $\lambda_{cyc} \in \{0, 0.1, 1, 10, 50, 100\}$. Additional results for the BART-base model on the GYAFC-family dataset can be found in Appendix.

In general, the values of the ref-BLEU and style accuracy metrics increase while increasing the value of λ_{cyc}. However, the ref-BLEU increase appears to be quite limited for large λ_{cyc} values, while accuracy still exhibits some room for improvement. Notably, disabling the cycle-consistency loss (i.e., setting $\lambda_{cyc} = 0$) results in a significant performance drop in terms of ref-BLEU (i.e., -4.2 compared to $\lambda_{cyc} = 0.1$). The performance drop is even more pronounced in terms of style accuracy (-6.1). These results confirm the importance of the cycle-consistency loss.

Pre-trained style classifier. In this ablation study, we aim to analyze the impact of the pre-trained style classifier. By enabling/disabling the classifier component, we introduce or eliminate the classifier loss contribution (see Equations 4 and 5). To ensure a complete overview of the classifier contribution, we averaged the evaluation metrics reported in Figure 3 across all the models trained on each dataset. The results reported in Figure 3 show that the introduction of the pre-trained classifier in the training process has a positive impact on all four evaluation metrics. In terms of BLEU scores, it achieves negligible improvements. However, the style classifier yields a +1.0 BLEU score improvement in the music domain of the GYAFC dataset. The limited effect on BLEU can be motivated by the fact that the pre-trained style classifier's objective is to improve the style

Table 10. Results on the GYAFC-family dataset with mixed style for different mixing ratios – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

mixing	model	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
0-100	CrossAlignment [34]	9.1	50.9	21.5	15.4	74.9	26.1	16.2
	MultiDecoder [5]	18.0	50.8	30.2	26.6	69.2	35.3	28.6
	CycleGAN latent T5 (large)	15.6	44.5	26.3	23.1	37.4	24.2	22.0
	CycleGAN T5 (large)	**97.7**	95.5	**96.6**	**96.6**	96.3	**97.0**	**97.0**
	Llama2-7B-Chat	92.0	**96.5**	94.2	94.2	**96.9**	94.4	94.4
25-75	CrossAlignment [34]	3.2	67.0	14.6	6.1	84.0	16.4	6.2
	MultiDecoder [5]	3.2	42.0	11.6	5.9	46.0	12.1	6.0
	CycleGAN latent T5 (large)	14.2	51.9	27.1	22.3	52.7	27.4	22.4
	CycleGAN T5 (large)	**62.2**	**98.3**	**78.2**	**76.2**	**95.8**	**77.2**	**75.4**
	Llama2-7B-Chat	58.4	97.0	75.3	72.9	92.6	73.5	71.6
33-66	CrossAlignment [34]	3.8	67.2	16.0	7.2	82.7	17.7	7.3
	MultiDecoder [5]	4.0	46.1	13.6	7.4	48.7	14.0	7.4
	CycleGAN latent T5 (large)	15.5	49.1	27.6	23.6	49.9	27.8	23.7
	CycleGAN T5 (large)	**65.6**	**97.3**	**79.9**	**78.3**	**94.3**	**78.7**	**77.4**
	Llama2-7B-Chat	58.1	97.3	75.2	72.8	93.9	73.9	71.8
50-50	CrossAlignment [34]	4.5	61.1	16.6	8.4	77.6	18.7	8.5
	MultiDecoder [5]	4.6	33.7	12.5	8.1	33.4	12.4	8.1
	CycleGAN latent T5 (large)	15.8	54.4	29.3	24.5	52.7	28.9	24.3
	CycleGAN T5 (large)	**68.7**	**92.3**	**79.6**	**78.8**	**91.0**	**79.1**	**78.3**
	Llama2-7B-Chat	63.7	90.4	75.9	74.7	85.9	74.0	73.2
66-33	CrossAlignment [34]	3.5	62.2	14.8	6.6	81.9	16.9	6.7
	MultiDecoder [5]	3.6	42.5	12.4	6.6	43.5	12.5	6.6
	CycleGAN latent T5 (large)	15.5	45.9	26.7	23.2	45.9	26.7	23.2
	CycleGAN T5 (large)	**68.2**	90.6	**78.6**	**77.8**	**91.7**	**79.1**	**78.2**
	Llama2-7B-Chat	66.9	**91.0**	78.0	77.1	85.2	75.5	74.9
75-25	CrossAlignment [34]	2.8	60.8	13.0	5.4	83.8	15.3	5.4
	MultiDecoder [5]	2.9	42.7	11.1	5.4	42.7	11.1	5.4
	CycleGAN latent T5 (large)	14.1	44.5	25.0	21.4	46.4	25.6	21.6
	CycleGAN T5 (large)	**64.2**	88.9	75.5	74.6	**93.4**	**77.4**	**76.1**
	Llama2-7B-Chat	63.5	**95.2**	**77.8**	**76.2**	86.9	74.3	73.4
100-0	CrossAlignment [34]	7.5	94.3	26.6	13.9	84.0	25.1	13.8
	MultiDecoder [5]	13.1	88.2	34.0	22.8	74.1	31.2	22.3
	CycleGAN latent T5 (large)	15.1	91.4	37.2	25.9	88.3	36.5	25.8
	CycleGAN T5 (large)	**92.1**	**97.0**	**94.5**	**94.5**	**95.5**	**93.8**	**93.8**
	Llama2-7B-Chat	91.7	93.3	92.5	92.5	90.7	91.2	91.2

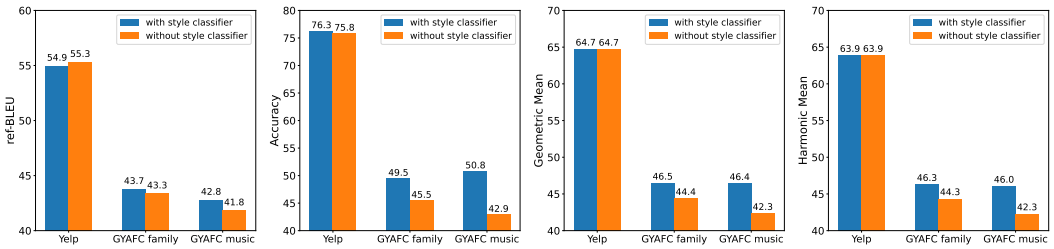

Fig. 3. Effect of the pre-trained style classifier on the evaluation metrics across different datasets. Results are averaged over all the tested models.

transfer accuracy, and thus, it does not necessarily affect the BLEU scores. On the contrary, we observe remarkable improvements in terms of style accuracy on the two domains of the GYAFC dataset. Specifically, we achieve an absolute gain of +4.0 and +7.9 points in accuracy scores, which corresponds to the relative improvements of +8.8% and +18.4% when compared to the classifier-free counterparts. Finally, by analyzing the impact on the geometric mean and harmonic mean of BLEU and style accuracy, we can observe an overall improvement of up to +4.1 and +3.7 points, respectively. The geometric mean and harmonic mean provide a more comprehensive evaluation of the overall performance of the approach, taking into account the trade-off between the two separate metrics. These results, therefore, confirm the effectiveness of the pre-trained classifier in enhancing the quality of the generated text.

The evaluation results show a surprising lack of performance improvement on the Yelp dataset. One possible explanation for this phenomenon is that the sentiment transfer task already achieves high style accuracy scores, even without the pre-trained classifier. This may suggest that the model's pre-existing capability to perform style transfer is already sufficient to achieve high accuracy scores, making the pre-trained classifier's contribution negligible in this case. Also, the larger size of the Yelp dataset may already provide the model with a sufficient amount of training data to effectively capture style transfer patterns. Moreover, as described in Section 5.3, since the Yelp dataset does not include human references for the validation set, style accuracy is already taken into account when performing the hyperparameter tuning and selecting the best checkpoints. This may be another possible explanation for the limited impact of the pre-trained classifier on this dataset.

As the quality of the pre-trained style classifier may affect the overall performance of our proposed architecture, we extend this ablation study by also testing other style classifiers. In addition to the BERT-base model used in our architecture, we test the following models: BERT-large, RoBERTa-base [20], RoBERTa-large, and DistilBERT-base. More detailed results for the BART-base model on the GYAFC-family dataset can be found in Appendix.

Overall, we observe that the specific style classifier chosen does not have a significant impact on performance. Specifically, the differences among the various classifiers in ref-BLEU are negligible (i.e., ± 0.1), and similarly for style accuracy, where fluctuations range from ± 0.4 to ± 2.5. The largest differences in performance are observed with the DistilBERT-base model, showing a drop of -1.0 and -4.6 in ref-BLEU and accuracy scores, respectively. This result is expected, given that the DistilBERT-base model is the lightest among those tested. Nevertheless, all tested models perform generally well, indicating that the quality of the pre-trained style classifier has a limited impact on the final performance.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a new approach to self-supervised Text Style Transfer using Cycle-consistent Generative Adversarial Networks (CycleGANs). Thanks to the joint use of cycle-consistency and a pre-trained style classification loss, our method is able to effectively transfer the style of a source text to a target text without the need for labeled data. The experimental results, achieved on three benchmark datasets and two different TST tasks, show that our method performs better than state-of-the-art approaches in terms of quality of the generated text and ability to preserve the content of the source text, particularly when mixed-style inputs are processed.

Limitations. The application of the proposed approach to real case studies should consider the following potential limitations. (1) We made the assumption that the source and target domains have approximately the same distributions. As a result, the model could be unable to learn the correct mapping between the two style attributes if this assumption does not hold true. (2) The presented approach may be misused to maliciously manipulate the text style or sentiment. For example, by transferring the style of credible sources to untruthful content, the proposed method

might be employed to automatically generate fake news or propaganda. (3) The currently proposed architecture handles one specific pair of source and target styles. This leads to scalability issues, as it would require training a separate architecture for each new pair of styles.

Despite the aforementioned limitations, the usability of the proposed method is quite promising in various real-world scenarios. With a responsible deployment and a careful consideration of the main ethical concerns, our approach can relevantly contribute to the advancement of the TST research field and enable innovative NLP applications in fields such as marketing, content generation, and digital storytelling.

Future work. We plan to extend our work across multiple directions. (1) We aim to expand the capabilities of the proposed architecture by investigating its performance in a multilingual setting. Transferring the style attributes across languages is potentially challenging as entails not only capturing stylistic nuances but also handling language-specific characteristics. By considering this aspect, we can evaluate the model's ability to generalize and adapt to diverse linguistic contexts. (2) The flexibility of our method allows us to explore its applicability to new domains and tasks. For instance, we would like to further explore the following two related tasks: *Aspect-level style transfer* [13], and *Controllable text generation* [10]. In both cases, the goal is to selectively transfer specific attributes or aspects of the writing style while preserving the rest. (3) We also envisage to extend our approach to handle more than a single pair of styles simultaneously (see Section 4.5).

ACKNOWLEDGMENTS

The work by Giuseppe Gallipoli was carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1551.11-10-2022, PE00000004). This study was also partially carried out within the FAIR (Future Artificial Intelligence Research) and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555.11-10-2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

REFERENCES

[1] Liqun Chen et al. 2018. Adversarial Text Generation via Feature-Mover's Distance. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Vol. 31. Curran Associates, Inc., 4671–4682. https://proceedings.neurips.cc/paper/2018/hash/074177d3eb6371e32c16c55a3b8f706b-Abstract.html

[2] Hyung Won Chung et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[3] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5997–6007. https://doi.org/10.18653/v1/P19-1601

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Zhenxin Fu et al. 2018. Style Transfer in Text: Exploration and Evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://doi.org/10.1609/aaai.v32i1.11330

[6] Ian Goodfellow et al. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc., Montréal. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[7] Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A Probabilistic Formulation of Unsupervised Text Style Transfer. In *8th International Conference on Learning Representations, ICLR 2020*. Addis Abeba.

https://arxiv.org/abs/2002.03912

[8] Zhiting Hu et al. 2018. Toward Controlled Generation of Text. arXiv:1703.00955 [cs.LG]

[9] Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. Text Style Transfer: A Review and Experimental Evaluation. *SIGKDD Explor.* 24, 1 (2022), 14–45. https://doi.org/10.1145/3544903.3544906

[10] Zhiting Hu and Li Erran Li. 2021. A Causal Lens for Controllable Text Generation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 24941–24955. https://proceedings.neurips.cc/paper/2021/hash/d0f5edad9ac19abed9e235c0fe0aa59f-Abstract.html

[11] Fei Huang et al. 2021. NAST: A Non-Autoregressive Generator with Word Alignment for Unsupervised Text Style Transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 1577–1590. https://doi.org/10.18653/v1/2021.findings-acl.138

[12] Yufang Huang et al. 2020. Cycle-Consistent Adversarial Autoencoders for Unsupervised Text Style Transfer. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2213–2223. https://doi.org/10.18653/v1/2020.coling-main.201

[13] Di Jin et al. 2022. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics* 48, 1 (March 2022), 155–205. https://doi.org/10.1162/coli_a_00426

[14] Nitish Shirish Keskar et al. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *CoRR* abs/1909.05858 (2019). http://arxiv.org/abs/1909.05858

[15] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751. https://doi.org/10.3115/v1/D14-1181

[16] Guillaume Lample et al. 2019. Multiple-Attribute Text Rewriting. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=H1g2NhC5KQ

[17] Mike Lewis et al. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[18] Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1865–1874. https://doi.org/10.18653/v1/N18-1169

[19] Dayiheng Liu et al. 2019. Revision in Continuous Space: Fine-Grained Control of Text Style Transfer. *CoRR* abs/1905.12304 (05 2019). https://arxiv.org/abs/1905.12304

[20] Yinhan Liu et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[21] Michela Lorandi et al. 2023. Adapting the CycleGAN architecture for text style transfer. (2023). https://doi.org/10.5281/zenodo.8268839

[22] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7

[23] Fuli Luo et al. 2019. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*. Macao. https://arxiv.org/abs/1905.10060

[24] Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and Match: Learning-free Controllable Text Generationusing Energy Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 401–415. https://doi.org/10.18653/v1/2022.acl-long.31

[25] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, Brussels, Belgium, 186–191. https://doi.org/10.18653/v1/W18-6319

[26] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style Transfer Through Back-Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 866–876. https://doi.org/10.18653/v1/P18-1080

[27] Colin Raffel et al. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[28] Sudha Rao and Joel Tetreault. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 129–140. https://doi.org/10.18653/v1/N18-1012

[29] Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein Editing for Unsupervised Text Style Transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 3932–3944. https://doi.org/10.18653/v1/2021.findings-acl.344

[30] Emily Reif et al. 2022. A Recipe for Arbitrary Text Style Transfer with Large Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, 837–848. https://doi.org/10.18653/v1/2022.acl-short.94

[31] Abhilasha Sancheti, Kundan Krishna, Balaji Vasan Srinivasan, and Anandhavelu Natarajan. 2020. Reinforced Rewards Framework for Text Style Transfer. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Proceedings, Part I*. Lisbon. https://arxiv.org/abs/2005.05256

[32] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[33] Mingyue Shang et al. 2019. Semi-supervised Text Style Transfer: Cross Projection in Latent Space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 4937–4946. https://doi.org/10.18653/v1/D19-1499

[34] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style Transfer from Non-Parallel Text by Cross-Alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California. https://arxiv.org/abs/1705.09655

[35] Yukai Shi et al. 2021. GTAE: Graph Transformer–Based Auto-Encoders for Linguistic-Constrained Text Style Transfer. *ACM Trans. Intell. Syst. Technol.* 12, 3, Article 32 (jun 2021), 16 pages. https://doi.org/10.1145/3448733

[36] Martina Toshevska and Sonja Gievska. 2022. A Review of Text Style Transfer Using Deep Learning. *IEEE Transactions on Artificial Intelligence* 3, 5 (2022), 669–684. https://doi.org/10.1109/TAI.2021.3115992

[37] Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[38] Ashish Vaswani et al. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[39] Yunli Wang et al. 2019. Harnessing Pre-Trained Neural Networks with Rules for Formality Style Transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3573–3578. https://doi.org/10.18653/v1/D19-1365

[40] Thomas Wolf et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[41] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2019. A Fast Proximal Point Method for Computing Exact Wasserstein Distance. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019 (Proceedings of Machine Learning Research, Vol. 115)*. AUAI Press, 433–453. http://proceedings.mlr.press/v115/xie20b.html

[42] Jingjing Xu et al. 2018. Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 979–988. https://doi.org/10.18653/v1/P18-1090

[43] Wei Xu et al. 2012. Paraphrasing for Style. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, Mumbai, India, 2899–2914. https://aclanthology.org/C12-1177

[44] Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018. Learning Sentiment Memories for Sentiment Modification without Parallel Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1103–1108. https://doi.org/10.18653/v1/D18-1138

[45] Zhirui Zhang et al. 2018. Style Transfer as Unsupervised Machine Translation. *CoRR* abs/1808.07894 (08 2018). https://arxiv.org/abs/1808.07894

[46] Yanpeng Zhao et al. 2018. Language Style Transfer from Sentences with Arbitrary Unknown Styles. *CoRR* abs/1808.04071 (2018). http://arxiv.org/abs/1808.04071

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE Computer Society, Venice, 2242–2251. https://doi.org/10.1109/ICCV.2017.244

APPENDIX

This document contains the following appendices:

A) Hyperparameter settings;
B) Additional results on formality transfer;
C) Additional results on mixed-style inputs;
D) Additional results on cycle-consistency loss;
E) Additional results on classifier-guided loss;
F) Qualitative examples.

A HYPERPARAMETER SETTINGS

In Table 11 we report the optimal hyperparameter values used throughout the experiments on both the GYAFC and Yelp datasets.

Table 11. Optimal hyperparameter configurations for each dataset and model used in experiments.

dataset	generator model	learning rate	λ_{cyc}
GYAFC-family	BART-base	$5 \cdot 10^{-5}$	10
	BART-large	$5 \cdot 10^{-5}$	
	T5-small	$5 \cdot 10^{-5}$	
	T5-base	$5 \cdot 10^{-5}$	
	T5-large	$5 \cdot 10^{-5}$	
GYAFC-music	BART-base	$5 \cdot 10^{-5}$	1
	BART-large	$1 \cdot 10^{-5}$	
	T5-small	$5 \cdot 10^{-5}$	
	T5-base	$5 \cdot 10^{-5}$	
	T5-large	$5 \cdot 10^{-5}$	
Yelp	BART-base	$1 \cdot 10^{-4}$	10
	BART-large	$1 \cdot 10^{-5}$	
	T5-small	$1 \cdot 10^{-3}$	
	T5-base	$1 \cdot 10^{-4}$	
	T5-large	$5 \cdot 10^{-5}$	

B ADDITIONAL RESULTS ON FORMALITY TRANSFER

In this section, we report the additional results for the formality transfer task. Specifically, Table 12 shows the detailed results on the GYAFC-family dataset in the informal-to-formal style transfer direction. Even when we restrict the analysis to a specific style transfer direction, our proposed approach achieves the best performance (i.e., +10.3 ref-BLEU).

Table 13 shows the detailed results on the GYAFC-music dataset in the informal-to-formal style transfer direction. Similar to the GYAFC-family domain, our method largely outperforms the other approaches (i.e., +34.1 ref-BLEU).

Table 14 includes the detailed results obtained by Llama2 [37] on the GYAFC-family and GYAFC-music datasets in the informal-to-formal style transfer direction. The best ref-BLEU performance is achieved for $k = 5$ and $k = 3$, respectively. However, our approach confirms its superior performance by a large margin (i.e., +18.6 and +13.1 ref-BLEU, respectively).

Table 12. Results on the GYAFC-family dataset | informal → formal – ref-BLEU (ref-B), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

	ref-B	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
RetrieveOnly [18]	4.3	**92.5**	19.9	8.2	7.5	5.7	5.5
BackTranslation [26]	5.5	6.4	5.9	5.9	8.3	6.8	6.6
StyleEmbedding [5]	12.0	10.8	11.4	11.4	0.0	0.0	0.0
MultiDecoder [5]	17.3	5.0	9.3	7.8	0.0	0.0	0.0
CrossAlignment [34]	8.1	45.1	19.1	13.7	5.7	6.8	6.7
UnpairedTranslation [42]	4.3	36.9	12.6	7.7	7.5	5.7	5.5
DeleteOnly [18]	35.1	6.1	14.6	10.4	0.2	2.6	0.4
DeleteAndRetrieve [18]	24.4	33.6	28.6	28.3	4.6	10.6	7.7
TemplateBased [18]	39.9	16.3	25.5	23.1	5.2	14.4	9.2
UnsupervisedMT [45]	37.9	59.3	47.4	46.2	3.8	12.0	6.9
DualRL [23]	51.6	28.7	38.5	36.9	0.8	6.4	1.6
NASTLatentLearn [11]	51.1	37.6	43.8	43.3	30.0	39.1	37.8
CycleGAN BART (base)	58.7	42.2	49.8	49.1	47.0	52.5	52.2
CycleGAN BART (large)	59.3	41.2	49.4	48.6	46.2	52.3	51.9
CycleGAN T5 (small)	54.3	28.7	39.5	37.6	34.8	43.5	42.4
CycleGAN T5 (base)	56.7	28.9	40.5	38.3	32.1	42.7	41.0
CycleGAN T5 (large)	**61.9**	49.2	**55.2**	**54.8**	**53.2**	**57.4**	**57.2**

Table 13. Results on the GYAFC-music dataset | informal → formal – ref-BLEU (ref-B), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy. * denotes results from the paper.

	ref-B	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
DeepLatent [7]	26.4	**77.8**	45.3	39.4	25.6	26.0	26.0
ReinfRewards* [31]	28.6	72.3	45.4	41.0	–	–	–
MixAndMatch* [24]	–	19.0	–	–	–	–	–
CycleGAN BART (base)	57.6	53.7	55.6	55.6	48.4	52.8	52.6
CycleGAN BART (large)	55.3	42.8	48.7	48.3	38.4	46.1	45.3
CycleGAN T5 (small)	51.6	34.4	42.1	41.3	29.5	39.0	37.5
CycleGAN T5 (base)	55.2	36.5	44.9	43.9	33.4	42.9	41.6
CycleGAN T5 (large)	**62.7**	67.1	**64.9**	**64.8**	**61.6**	**62.1**	**62.1**

C ADDITIONAL RESULTS ON MIXED-STYLE INPUTS

Table 15 reports the results for the GYAFC-music formality transfer dataset in the mixed-style scenario. Our approach demonstrates superior performance to the other competitors in terms of ref-BLEU, geometric and harmonic means across all mixing ratios. Notably, the ref-BLEU performance gap with the second-best performer (i.e., Llama2) ranges from +8.4 in the 25 – 75 case to +22.2 in the 50 – 50 case. These results, similar to those observed in the GYAFC-family dataset, underscore the enhanced capability of our method in handling mixed-style texts, especially those without a predominant style.

Table 14. Results on the GYAFC-family and GYAFC-music datasets | informal → formal of Llama2-7B-Chat model for varying number of examples k in the 0/few-shot setting – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

dataset	k	ref-B	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
GYAFC-family	0	23.1	81.8	43.4	36.0	84.2	44.1	36.2
	1	20.9	90.8	43.5	34.0	92.2	43.9	34.1
	3	41.7	90.8	61.5	57.1	91.3	61.7	57.2
	5	**43.3**	89.5	**62.2**	**58.3**	90.1	**62.4**	**58.5**
	10	35.9	**92.6**	57.6	51.7	**93.1**	57.8	51.8
GYAFC-music	0	27.5	87.4	49.0	41.8	80.4	47.0	41.0
	1	39.2	92.2	60.1	55.0	87.4	58.5	54.1
	3	**49.6**	91.0	**67.2**	**64.2**	87.0	**65.7**	**63.2**
	5	44.7	91.9	64.1	60.1	87.9	62.7	59.2
	10	46.2	**93.0**	65.5	61.7	**88.9**	64.1	60.8

D ADDITIONAL RESULTS ON CYCLE-CONSISTENCY LOSS

Table 16 reports the detailed results of the ablation study on the effect of the λ_{cyc} hyperparameter. Notably, the best performance in terms of ref-BLEU and style accuracy is achieved for higher λ_{cyc} values. This underscores the importance of the cycle-consistency loss and emphasizes the need for accurate tuning of the corresponding scaling factor.

E ADDITIONAL RESULTS ON CLASSIFIER-GUIDED LOSS

In this section, we report in Tables 17-21 the detailed set of results of the ablation study analyzing the effect of the pre-trained style classifier. Additionally, the results of the ablation study on the model used as pre-trained style classifier are presented in Table 22.

More specifically, Tables 17 and 18 report the results achieved on the GYAFC-family dataset. Almost all the tested models benefit from the introduction of the pre-trained classifier. Notably, the best-performing model (T5 large) achieves an improvement up to +1.0 and +10.3 points in the BLEU score and style accuracy, respectively. The accuracy improvement is even higher while considering only the informal-to-formal direction (i.e., +16.9 points). Similar observations hold for the results reported in Tables 19 and 20 (e.g., +27.5 points in accuracy) which display the results on the music domain. Finally, Table 21 shows the results on the Yelp dataset. Although the best BLEU and accuracy scores are achieved by the models trained without the style classifier, it can be noticed that the introduction of the classifier-guided loss yields an improvement in the overall performance score, as indicated by the geometric mean and harmonic mean.

F QUALITATIVE EXAMPLES

In this section, we show qualitative examples of pairs of original and transformed texts.

To provide a more comprehensive evaluation of the quality of the generated text, we conduct a qualitative analysis by comparing it with both the ground truth and a subset of the baseline methods. This analysis allows us to identify the success and failure cases of our approach more tangibly and compare them with the outputs of other state-of-the-art methods. Tables 23 and 25 report a selection of success and failure cases for the formality transfer task within the family domain, Tables 24 and 26 contain the results in the music domain, whereas Tables 27 and 28 report qualitative results for the sentiment transfer task.

Table 15. Results on the GYAFC-music dataset with mixed style for different mixing ratios – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

mixing	model	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
0-100	CrossAlignment [34]	8.6	49.9	20.7	14.7	52.9	21.3	14.8
	MultiDecoder [5]	13.2	65.4	29.4	22.0	71.3	30.7	22.3
	CycleGAN latent T5 (large)	16.8	69.1	34.1	27.0	28.3	21.8	21.1
	CycleGAN T5 (large)	**97.9**	98.7	**98.3**	**98.3**	95.7	**96.8**	**96.8**
	Llama2-7B-Chat	80.4	**98.9**	89.2	88.7	**97.1**	88.4	88.0
25-75	CrossAlignment [34]	2.7	66.2	13.4	5.2	61.2	12.9	5.2
	MultiDecoder [5]	2.9	54.8	12.6	5.5	55.3	12.7	5.5
	CycleGAN latent T5 (large)	13.9	52.9	27.1	22.0	51.0	26.6	21.8
	CycleGAN T5 (large)	**61.1**	93.9	**75.7**	**74.0**	**95.6**	**76.4**	**74.6**
	Llama2-7B-Chat	52.7	**97.1**	71.5	68.3	92.0	69.6	67.0
33-66	CrossAlignment [34]	3.2	65.3	14.5	6.1	61.1	14.0	6.1
	MultiDecoder [5]	3.5	51.7	13.5	6.6	51.9	13.5	6.6
	CycleGAN latent T5 (large)	16.3	51.0	28.8	24.7	49.8	28.5	24.6
	CycleGAN T5 (large)	**65.5**	**94.3**	**78.6**	**77.3**	**95.4**	**79.0**	**77.7**
	Llama2-7B-Chat	49.0	93.4	67.7	64.3	89.7	66.3	63.4
50-50	CrossAlignment [34]	3.8	63.4	15.5	7.2	62.8	15.4	7.2
	MultiDecoder [5]	4.1	35.7	12.1	7.4	35.5	12.1	7.4
	CycleGAN latent T5 (large)	16.2	60.2	31.2	25.5	53.6	29.5	24.9
	CycleGAN T5 (large)	**69.2**	**94.3**	**80.8**	**79.8**	**93.0**	**80.2**	**79.4**
	Llama2-7B-Chat	46.9	89.4	64.8	61.5	86.2	63.6	60.7
66-33	CrossAlignment [34]	3.2	62.9	14.2	6.1	60.5	13.9	6.1
	MultiDecoder [5]	3.2	47.4	12.3	6.0	48.8	12.5	6.0
	CycleGAN latent T5 (large)	15.1	46.8	26.6	22.8	45.9	26.3	22.7
	CycleGAN T5 (large)	**69.7**	96.7	**82.1**	**81.0**	**95.2**	**81.5**	**80.5**
	Llama2-7B-Chat	56.2	**96.9**	73.8	71.1	94.0	72.7	70.3
75-25	CrossAlignment [34]	2.7	60.9	12.8	5.2	57.8	12.5	5.2
	MultiDecoder [5]	2.5	49.2	11.1	4.8	52.4	11.4	4.8
	CycleGAN latent T5 (large)	13.9	46.9	25.5	21.4	46.3	25.4	21.4
	CycleGAN T5 (large)	**68.2**	**97.5**	**81.5**	**80.3**	**94.6**	**80.3**	**79.3**
	Llama2-7B-Chat	56.1	94.1	72.7	70.3	91.5	71.6	69.6
100-0	CrossAlignment [34]	6.4	**96.7**	24.9	12.0	95.8	24.8	12.0
	MultiDecoder [5]	9.1	78.1	26.7	16.3	73.1	25.8	16.2
	CycleGAN latent T5 (large)	15.5	82.9	35.8	26.1	90.8	37.5	26.5
	CycleGAN T5 (large)	**92.1**	94.8	**93.4**	**93.4**	**95.8**	**93.9**	**93.9**
	Llama2-7B-Chat	83.4	89.6	86.4	86.4	90.7	87.0	86.9

In our qualitative analysis, we identify the main failure cases and summarize them below:

- *Metaphoric language:* the model's limited ability to accurately recognize and modify metaphoric language and idiomatic expressions. This can result in the model retaining the original expressions in the generated text, which may not conform to the desired style.
- *Slang:* the model's difficulty in accurately recognizing and modifying common words used with their slang meaning, particularly in cases where the conversion is from informal to formal language. In such cases, the model may consider the slang word as already formal and fail to convert it, resulting in the retention of the original expression.

Table 16. Effect of the λ_{cyc} hyperparameter without classifier-guided loss on the GYAFC-family dataset with BART (base) model – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

λ_{cyc}	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
0	39.0	6.1	15.4	10.6	6.8	16.3	11.6
0.1	43.2	46.2	44.7	44.7	45.2	44.2	44.2
1	43.1	47.6	45.3	45.2	46.5	44.8	44.7
10	42.8	44.0	43.4	43.4	43.1	42.9	42.9
50	43.5	47.4	45.4	45.4	46.6	45.0	45.0
100	**43.5**	**50.1**	**46.7**	**46.6**	**49.6**	**46.5**	**46.4**

Table 17. Effect of the classifier-guided loss on the GYAFC-family dataset – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
CycleGAN BART (base)	43.7	50.7	47.1	46.9	49.4	46.5	46.4
w/o style classifier	42.8	44.0	43.4	43.4	43.1	42.9	42.9
CycleGAN BART (large)	43.5	50.8	47.0	46.9	49.9	46.6	46.5
w/o style classifier	43.4	47.9	45.6	45.5	47.6	45.5	45.4
CycleGAN T5 (small)	42.1	38.7	40.4	40.3	39.6	40.8	40.8
w/o style classifier	42.1	39.2	40.6	40.6	39.9	41.0	41.0
CycleGAN T5 (base)	44.0	47.7	45.8	45.8	46.2	45.1	45.1
w/o style classifier	44.0	47.1	45.5	45.5	45.7	44.8	44.8
CycleGAN T5 (large)	**45.4**	**59.5**	**52.0**	**51.5**	**58.1**	**51.4**	**51.0**
w/o style classifier	44.4	49.2	46.7	46.7	47.9	46.1	46.1

Table 18. Effect of the classifier-guided loss on the GYAFC-family dataset | informal → formal – ref-BLEU (ref-B), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

	ref-B	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
CycleGAN BART (base)	58.7	42.2	49.8	49.1	47.0	52.5	52.2
w/o style classifier	57.2	34.5	44.4	43.0	38.4	46.9	46.0
CycleGAN BART (large)	59.3	41.2	49.4	48.6	46.2	52.3	51.9
w/o style classifier	59.9	47.8	53.5	53.2	**54.4**	57.1	57.0
CycleGAN T5 (small)	54.3	28.7	39.5	37.6	34.8	43.5	42.4
w/o style classifier	54.3	29.1	39.8	37.9	35.3	43.8	42.8
CycleGAN T5 (base)	56.7	28.9	40.5	38.3	32.1	42.7	41.0
w/o style classifier	57.7	32.3	43.2	41.4	36.3	45.8	44.6
CycleGAN T5 (large)	**61.9**	**49.2**	**55.2**	**54.8**	53.2	**57.4**	**57.2**
w/o style classifier	58.0	32.3	43.3	41.5	37.2	46.4	45.3

- *In-depth rephrasing:* the model's inability to perform more profound rephrasing of the input sentence when necessary to achieve the desired style transfer. This is a common limitation of non-parallel TST approaches, where the lack of parallel training data makes it challenging to learn more complex mappings between styles.

Table 19. Effect of the classifier-guided loss on the GYAFC-music dataset – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy. * denotes results from the paper.

	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
CycleGAN BART (base)	43.6	57.2	49.9	49.5	57.8	50.2	49.7
w/o style classifier	42.3	49.0	45.5	45.4	50.6	46.3	46.1
CycleGAN BART (large)	42.0	43.1	42.5	42.5	43.8	42.9	42.9
w/o style classifier	40.8	41.8	41.3	41.3	43.0	41.9	41.9
CycleGAN T5 (small)	40.6	37.9	39.2	39.2	39.0	39.8	39.8
w/o style classifier	40.6	37.6	39.1	39.0	38.6	39.6	39.6
CycleGAN T5 (base)	42.0	45.4	43.7	43.6	47.7	44.8	44.7
w/o style classifier	42.0	43.2	42.6	42.6	45.5	43.7	43.7
CycleGAN T5 (large)	**45.6**	**70.5**	**56.7**	**55.4**	**70.1**	**56.5**	**55.3**
w/o style classifier	43.3	43.0	43.1	43.1	45.1	44.2	44.2

Table 20. Effect of the classifier-guided loss on the GYAFC-music dataset | informal → formal – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

	ref-B	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
CycleGAN BART (base)	57.6	53.7	55.6	55.6	48.4	52.8	52.6
w/o style classifier	55.5	43.8	49.3	49.0	39.8	47.0	46.4
CycleGAN BART (large)	55.3	42.8	48.7	48.3	38.4	46.1	45.3
w/o style classifier	52.3	37.2	44.1	43.5	32.9	41.5	40.4
CycleGAN T5 (small)	51.6	34.4	42.1	41.3	29.5	39.0	37.5
w/o style classifier	51.7	35.5	42.8	42.1	29.8	39.3	37.8
CycleGAN T5 (base)	55.2	36.5	44.9	43.9	33.4	42.9	41.6
w/o style classifier	54.9	35.7	44.3	43.3	32.5	42.2	40.8
CycleGAN T5 (large)	**62.7**	**67.1**	**64.9**	**64.8**	**61.6**	**62.1**	**62.1**
w/o style classifier	58.3	48.5	53.2	53.0	44.4	50.9	50.4

Table 21. Effect of the classifier-guided loss on the Yelp dataset – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
CycleGAN BART (base)	55.7	78.8	**66.3**	**65.3**	77.8	**65.8**	**64.9**
w/o style classifier	53.1	**82.7**	66.3	64.7	**81.3**	65.7	64.2
CycleGAN BART (large)	56.5	75.1	65.1	64.5	74.6	64.9	64.3
w/o style classifier	**56.9**	73.9	64.8	64.3	73.1	64.5	64.0
CycleGAN T5 (small)	53.0	78.0	64.3	63.1	78.2	64.4	63.2
w/o style classifier	54.4	76.5	64.5	63.6	77.8	65.1	64.0
CycleGAN T5 (base)	54.2	76.6	64.4	63.5	77.3	64.7	63.7
w/o style classifier	55.4	74.2	64.1	63.4	74.7	64.3	63.6
CycleGAN T5 (large)	55.3	72.9	63.5	62.9	73.7	63.8	63.2
w/o style classifier	56.6	71.9	63.8	63.3	71.9	63.8	63.3

Table 22. Effect of different pre-trained style classifier models on the GYAFC-family dataset with BART (base) model – ref-BLEU (ref-B avg), style accuracy with BERT and TextCNN (acc_{BERT}, acc_{CNN}), geometric mean (GM) and harmonic mean (HM) of ref-BLEU and style accuracy.

style classifier	ref-B avg	acc_{BERT}	GM	HM	acc_{CNN}	GM	HM
BERT-base	**43.7**	**50.7**	**47.1**	**46.9**	**49.4**	**46.5**	**46.4**
BERT-large	43.6	48.2	45.8	45.8	47.7	45.6	45.6
RoBERTa-base	43.6	49.1	46.3	46.2	47.9	45.7	45.7
RoBERTa-large	43.6	50.3	46.8	46.7	49.2	46.3	46.2
DistilBERT-base	42.7	46.1	44.4	44.3	44.7	43.7	43.7

A more detailed analysis of the qualitative results follows.

F.1 Formality transfer

Comparison with the reference annotations. Tables 23 and 24 include success and failure cases for both directions of the formality transfer task, for both the family and music domains, respectively. Considering the family domain, the first two examples in Table 23 shows the ability of the model to transform an informal text into the corresponding formal version. Specifically, in the first case the model capitalizes the first letter of the sentence, the contracted and colloquial form "dont" is converted into "do not", the full stop is added at the end of the text and the slang terms "dat" and "da" are transformed into their proper versions "that" and "the", respectively. Although two occurrences of the word "dat" are present in the input document, to avoid repetitions the model does not replace both of them with "that" but one of the two is mapped to "who", thus denoting both language variety in the generated text and the model's understanding capabilities to recognize nuanced differences in language use. In the second case, the model correctly introduces the proper subject in the output sentence and replaces the smiling emoticon with a full stop, resulting in a perfect match with one of the available references. Considering the formal-to-informal direction, the model transforms the first letter of the sentence into its lowercase version, replaces the subject "you" with its contracted form "u" and rephrases the wording "in that manner" with the more informal version "like that". In the second example, similar transformations are applied by the model (i.e., lowercase for the first letter, "you" → "u"); moreover, "do not" is contracted into "dont", "are" is replaced by the informal abbreviation "r" and punctuation is removed from the sentence. Although the generated text does not match any of the references, it is successfully transferred to the informal style.

Considering now the potential failure cases reported in the bottom part of Table 23, we can observe that in one example the model successfully capitalizes the first letter and adds a full stop at the end of the sentence, but fails to modify the informal language in the remaining text. This is probably caused by the use of the metaphoric expression "hit the nail on the head" which the model may not recognize as informal language and, therefore, fails to modify. In the second example, the model successfully applies some modifications to improve the formality of the input text, such as capitalizing the first letter and replacing "ur" with "you are". However, the model fails to recognize the slang term "hoe" and, therefore, copies it to the output text, likely because it is interpreted in its literal meaning. This example highlights a limitation of the model in accurately recognizing and modifying slang words, which can result in the retention of inappropriate language in the generated text. Considering both failure cases in the formal-to-informal direction, the model introduces informal elements in the text (i.e., lowercase, contractions, punctuation removal) but the generated sentences do not match the corresponding references. This limitation in accurately

modifying informal language may be due to the examples' complexity, where significant rephrasing is required to achieve the desired style transfer. Such cases can be particularly challenging for non-parallel TST approaches, where the lack of parallel training data makes it difficult to learn complex mappings between styles.

Table 24 reports a selection of success and failure cases for both directions of the formality transfer task within the music domain. In the first two examples, the model correctly transforms the informal samples into their formal counterparts. More precisely, in the first case several modifications are applied: the first letter is capitalized, abbreviations are mapped to their extended versions (i.e., "2" → "to", "ur" → "your", "u" → "you") and three exclamation marks are replaced with a full stop. In the second example, in addition to similar modifications performed in previous cases, the model corrects the spelling of the word "like"; moreover, the generated text adopts the proper capitalization while the source sentence is entirely written in uppercase. This results in an almost perfect match with one of the proposed references. Considering the formal-to-informal direction, the model correctly uses contractions, abbreviations and no capitalization when rewriting the input text in informal style. From the second example, it is also possible to see that the model replaces the words "recall" and "television" with the more common alternatives "remember" and "tv", respectively.

In the first failure case, the model tries to convert the source text into its corresponding formal version by capitalizing the first letter and adding a punctuation mark at the end of the sentence but it is not able to correct and substitute the words "no" and "sight" with their homophones "know" and "site", respectively. In the second example, the generated text closely resembles a copy of the input sentence. This can be attributed to the unconventional formatting of the word "respect" which is written in uppercase letters with each letter separated by a dot. Such formatting may have caused the model to interpret it as an acronym or a specific entity, making it challenging to effectively transform and generate the desired output. In the first failure case of the informal-to-formal direction, the generated sentence displays certain features commonly seen in informal texts, such as the absence of a subject and contracted forms. However, it does not correspond to any of the provided references. It is worth noting that the model fails to recognize and modify the expression "under the weather", which likely contributes to the discrepancy between the generated output and the desired reference sentences. It is possible that the model did not understand the idiom "under the weather", which typically refers to someone feeling unwell or sick. Finally, in the last example, the model nearly replicates the input text without substantial modifications. Once again, a more elaborate rewriting would be necessary to achieve a closer match with the proposed references.

Comparison with the baseline methods. Tables 25 and 26 report several examples of generated outputs for the formality transfer task for both family and music domains, respectively. All baseline methods struggle with preserving the source text content while converting style attributes from informal to formal and vice versa (see Table 25). Conversely, our approach achieves better preservation of the original content. According to the quantitative results reported in Section 5.5, the formality transfer task is more challenging than the sentiment transfer task, especially for content preservation. As an example, in the second case of the informal-to-formal direction from the family domain, only two out of five baselines include the word "avatar" in the generated sentence while the outputs of the remaining three practically have no overlap with the source text. On the contrary, our model perfectly matches one of the proposed references. By incorporating cycle-consistency losses, the model is encouraged to maintain a certain level of coherence and consistency in the generated text while effectively transforming style attributes. This aspect contributes to the better preservation of the original content, addressing a common challenge faced by baseline methods.

Table 23. Examples from GYAFC-family test set – success and failure cases.

		informal → formal	formal → informal
success	Source	i dont like a guy dat does dat all da time	Do you feel in that manner?
	Generated	I do not like a guy who does that all the time.	do u feel like that?
	References	I do not like men that does that all of the time. I do not find a man appealing who behaves that way. I don't want a man who does that. I do not like a guy that does that all of the time.	do u fell that way ? how u feel Is that how you feel? is that how u feel?
	Source	Sounds like a rhetorical question :)	I do not know, perhaps you are young.
	Generated	It sounds like a rhetorical question.	i dont know perhaps u r young
	References	It sounds like a rhetorical question. That sounds like a rhetorical question. That sounds like a rhetorical question. Are you posing a rhetorical question?	Maybe your young, I don't know. maybe your young... IDK. MAYBE AGE IS AN ISSUE maybe you're just young, idk
failure	Source	bembie hit the nail on the head	I can lift him.
	Generated	Bembie hit the nail on the head.	I can lift him :)
	References	You are speaking the truth. Bembie got the answer correct. Bembie got it right. Bembie reached the proper conclusion.	yup, i can lift him, he's only 185....lol i can lift him no problem Yep I can lift him. He's a weakling, I can pick him up!
	Source	so in my opinon ur not a hoe!	It is a very long list.
	Generated	So in my opinion you are not a hoe!	its a very long list
	References	Therefore, in my opinion, you are not promiscuous. In my opinion you are not a hoe. So in my opinion you are not a promiscuous woman. In my opinion, you are not promiscuous.	the list could go on and on. its a long ass list That list could go on forever the list is endless, tbh

Table 24. Examples from GYAFC-music test set – success and failure cases.

		informal → formal	formal → informal
success	Source	explain 2 ur parents that u really want 2 act !!!	Can you not rhyme the words together?
	Generated	Explain to your parents that you really want to act.	cant u rhyme the words together?
	References	Explain to your parents that you want to act. Explain to your parents that you very much want to act. You should explain to your parents your passion is acting. Explain to your parents that you truly want to act.	Cant you just rhyme the words with themselves? Can ya' not rhyme da damn wo'ds togeder? Can u rhyme the words? can't you just rhyme the words???
	Source	I LKE THEM ALSO THEY ARE SOOOO CUTE!	I cannot recall the last time I turned on my television.
	Generated	I like them. Also, they are so cute!	i cant remember the last time i turned on my tv
	References	I like them; also, they are adorable. I like them, also. They are so cute. I like them because they are cute. I like them as well, they are so cute!	I can't remember when I last turned on my TV. I don't remember when I turned my TV on. i dont remember when i watched tv last I dont even remember the last time i watched tv
failure	Source	does anybody no a web sight for a casting call	I am unsure, as I am currently under the weather.
	Generated	Does anybody no a web sight for a casting call?	unsure, as im currently under the weather.
	References	Does anyone know a website for a casting call? What website has casting calls? Is anyone aware of a "Casting Call" website? Are there any web sites for casting calls?	Umm..., don't know..., my head hurts... I dunno, I'm sick, dude I don't know as I'm feeling sick. idk... i don't feel good right now.
	Source	He just needs to R.E.S.P.E.C.T... me my pplz... lol	I can't listen.
	Generated	He just needs to R.E.S.P.E.C.T. me my pplz. Lol.	I cant listen.
	References	He needs to respect me and my people. He needs to show respect to me and my friends. He should respect myself and my colleagues. he just need to respect me and my people.	I turned my ears away ! Not listening..... I didn't want to hear it La la la I can't hear you!

DeepLatent [7] is the only baseline in the music domain whose outputs are publicly available. In almost all four reported examples, it generates sentences significantly different from the source text (see Table 26). This results in poor performance in the preservation of the original meaning, as confirmed also by the much lower BLEU scores with respect to our model.

F.2 Sentiment transfer

Comparison with the reference annotations. We analyze sentiment transfer cases in both directions (see Table 27). Considering the first negative-to-positive success case, the model not only removes

Table 25. Examples from GYAFC-family test set – comparison with baselines.

	informal → formal	formal → informal
Source	i guess its normal, but id just move on	I do not think you are ready if you have to ask other people.
References	I suppose it is normal, but I would just move on. I imagine it's common, however, I'd simply proceed. I guess that it is normal, but I would move on. I guess it is normal but I would just move on.	if u have to ask other people i really dont think ur ready to If u ask others, I don't think ur ready to you ain ready if u gotta ask the folks youre not ready if u need to ask others
RetrieveOnly [18]	it typically on from there .	but i ' m and think thats - maybe i am a .
CrossAlignment [34]	i would not care about them and they are attractive .	i would tell him if you just want to get out with him
UnsupervisedMT [45]	i am not normal , but id just move on to the club .	i do n't think you are ready if you have to ask other people it is .
DualRL [23]	i guess it is normal , but just move on .	i dont think you are ready if you have to ask other people .
NASTLatentLearn [11]	i guess it normal , but i just do on	i do not think you are ready if you have to ask other people .
Ours	I guess it is normal, but I would just move on.	i dont think ur ready if u have to ask other people
Source	r u talking about ur avatar?	I apologize, but I do not know. I wish the both of you luck.
References	Are you referring to your avatar? Are you talking about your avatar? Are you talking about your avatar? Are you talking about your avatar?	hey im sorry i don't know and i wish best of luck to both of you! Im sorry, idk.i wish u both luck sorry i dont know, good luck to you both! sorry, but idk. I still wish you the best.
RetrieveOnly [18]	it is not pleasant driving a - 100 .	yes , your on the wrong .
CrossAlignment [34]	you should be happy with marriage .	i think about the same thing i would know what you like me .
UnsupervisedMT [45]	near you talking about passion , without even if it .	i apologize , but i do n't know i wish you the both of you hookin
DualRL [23]	it is talking about avatar avatar ?	i dunno er
NASTLatentLearn [11]	do you talking about it avatar ,	i do , but i do not know ... i wish the both of you luck .
Ours	Are you talking about your avatar?	sorry but i dont know i wish the both of you luck

Table 26. Examples from GYAFC-music test set – comparison with baselines.

	informal → formal	formal → informal
Source	im a huge green day fan!!!!!	How old are you? You should be at least 18 years old.
References	I am a huge fan of the band Green Day. I am a big fan of the band Green Day. I am a big fan of Green Day! I am a huge Green Day fan.	i dont know how old ru?ur supposed to be 18 ur supposed to be 18, how old r u? Are you over 18? How old r u? how old is ur age? u must be at least 18
DeepLatent [7]	I am a fan of the movie .	How old are you ... you should be at least 18 years old .
Ours	I am a huge Green Day fan.	how old r u u should be at least 18 years old
Source	YOUR WASTING YOUR TIME, SAYS THE BOY.	It is not close to a PlayStation Portable (PSP), but it is close enough.
References	The boy said you are wasting your time. The boy says you're wasting your time. The boys said, "You are wasting your time". "You are wasting your time", says the boy.	Not close to a PSP, but close enough. ITS NOT A PSP BUT ITS YOURS Eh its not exactly a PSP but its good enough it isn't close to a PSP but its close enuf
DeepLatent [7]	Try the fourth of Narnia , but I am not sure .	i think its going to get a goddamn door (but it is easy to get enough .
Ours	"You are wasting your time", says the boy.	its not close to a psp but its close enough

the negation "not" but also strengthens the positive sentiment by adding the word "definitely", as done by the first reference. In the second case, the model is able to modify two aspects of the input sentence, therefore confirming its ability to deal with different aspects at the same time. Considering the positive-to-negative direction, the first example shows that the model correctly modifies both negative adjectives converting them into their positive counterparts. According to the second example, it can be seen that, in addition to modifying the sentiment of the input text, the model shows solid language understanding capabilities: instead of simply adding "limited", it also replaces "variety" with "number". Although this may be suboptimal in terms of BLEU score, language diversification is generally a positive feature. In all the reported success cases, it can be noticed that the model not only correctly modifies the sentiment of the source sentence but does it in a consistent way from a semantic point of view (e.g., over fried → perfectly fried, hard → soft, amazing → disgusting).

Moving to the failure cases, in the first example, the model almost copies the input sentence by adding the word "good" but the expression "run of the mill" is kept unchanged. The same pattern can be observed in the positive-to-negative case where the source sentence is copied to the output

Table 27. Examples from Yelp test set – success and failure cases.

		negative → positive	positive → negative
success	Source	we will not be coming back .	food is very fresh and amazing !
	Generated	we will definitely be coming back.	food is very bland and disgusting!
	References	we will definitely be coming back ! we will be back soon, we will come back soon we will quickly come back	food was old and stale food is not fresh and average the food is very bad and disgusting the food is not good and disgusting !
	Source	it was over fried and very hard .	the variety of sushi rolls makes for a good eating .
	Generated	it was perfectly fried and very soft.	the limited number of sushi rolls makes for a bad eating.
	References	it was a little overcooked it was cooked at its point and very soft , exquisite it was not over fried and not hard, it was fried just right and very tender	there is limited variety for sushi rolls . the variety of sushi roll does not tasted good no variety of sushi rolls makes for a no good eating . the variety of sushi rolls make for a bad eating
failure	Source	even the ice cream was very , very run of the mill .	the soup is absolutely to die for .
	Generated	even the ice cream was very good, very run of the mill.	the soup is absolutely to die for.
	References	even the ice cream was very , very unique and good . the ice cream is very nice . the ice cream is good the ice cream is very perfect .	the soup will make you want to die the soup is dreadful . the soup is bad . the soup is not good
	Source	what about the time i wasted ?	bartender made a great dirty martini !
	Generated	what about the time i saved?	bartender made a dirty martini!
	References	service was quick and swift i really enjoy my time this is really worth it i enjoy my time .	bartender gave me a watered down martini bartender made a bad martini bartener made a nasty dirty martini bartender made a bad dirty martini !

without modification. This is probably due to the use of the expression "to die for", which may be erroneously recognized as already negative and therefore is not modified. Similarly to the formality transfer task, these examples highlight a challenge faced by our model when dealing with texts that include idiomatic expressions. The model struggles to correctly identify and handle these expressions, leading to failure cases where the sentiment transfer is not accurately achieved. In the second negative-to-positive failure case, the model attempted to transform the word "wasted" into "saved", resulting in a shift towards a positive nuance. However, it is important to note that this generated output does not align with any of the provided references. The discrepancy between the model's output and the references can be attributed to the fact that the desired sentiment expression in the references may require a more significant rephrasing of the input sentence. Since our approach is trained in a self-supervised setting, where explicit supervision for specific rephrasing patterns is not provided, achieving a closer match to the references in such cases becomes more challenging. Lastly, the final example demonstrates another limitation of the model, where it fails to recognize the term "dirty martini" and, similar to a previous case, incorrectly assumes a negative sentiment in the sentence. Consequently, the model does not explicitly modify the sentiment of the text but only removes the adjective "great". This indicates that the model's performance is hindered when encountering domain-specific terms or expressions that are not adequately identified and processed.

Comparison with the baseline methods. When comparing our approach with the baseline methods, it is observed that the majority of the baselines achieve successful sentiment transfer. However, similarly to the formality transfer task, they struggle to preserve the original content of the text, as indicated in Table 28. Let us consider, as an example, the second case reported for the negative-to-positive direction. Our model perfectly matches one of the available references whereas the baselines either do not transfer the style (e.g., DualRL [23]) or modify the text content (e.g., GTAE [35], NASTLatentLearn [11]). Considering the first example in the positive-to-negative case, all the baselines considered keep the adjective "hot" unchanged and two of them also do not modify the positive adverb "perfectly", whereas our model correctly transfers the sentiment of the entire

Table 28. Examples from Yelp test set – comparison with baselines.

	negative → positive	positive → negative
Source	if i could give zero stars i def would .	it 's hot , cooked perfectly , and delicious !
References	the stars was 5 plus	is was horribly cooked and bland
	if i could give 5+ stars and is great	it 's cold , cooked imperfectly , and bad taste .
	if i could give ten stars , i would definitely do it	it 's cold , cooked unperfectly , and suck !
	if i could give 5+ stars i def would	it 's cold , not cooked perfectly , and taste bad
RetrieveOnly [18]	best part is , everything is made from scratch and you .	maybe the hot dog is cold , but the chili is hot .
DualRL [23]	if i could give it a def would def recommend it .	it 's hot , over cooked , and cold !
GTAE [35]	if i could give perfect stars i def deliciously .	it 's hot , cooked terrible , and disappointing !
NASTLatentLearn [11]	if i could give a stars i def would .	it 's hot , cooked perfectly , and bland !
MixAndMatch [24]	if i could give him stars i definitely would .	it ' s hot , cooked perfectly , and horrible !
Ours	if i could give ten stars i def would.	it's cold, overcooked, and bland!
Source	tasted really old , i could n't believe it .	this place is super yummy !
References	tasted really fresh , i could n't believe it .	this place is super yucky !
	tasted really fresh , i could n't believe it	this place is not yummy at all !
	tasted really new , i could n't believe it	this place is terrible !
	very new taste , it is believable	this place is lacking in taste
RetrieveOnly [18]	really really really strong margaritas !	i would give this restaurant a zero , if that was an option .
DualRL [23]	tasted really old , i could definitely believe it .	this place is super yummy ?
GTAE [35]	tasted really genuine , i could deliciously wonderfully it .	this place is super worst !
NASTLatentLearn [11]	i really good , i could it believe it .	this place is terribly boring !
MixAndMatch [24]	tasted really amazing , i could n ' t believe it .	this place is so murky !
Ours	tasted really fresh, i couldn't believe it.	this place is super yuck!

sentence. Finally, in the last example, our model is the only one that properly converts "yummy" into "yuck". This confirms the ability of our model to transfer the style consistently from a semantic point of view, as observed in the analysis of success cases.