TinyStyler: Efficient Few-Shot Text Style Transfer with Authorship Embeddings

Zachary Horvitz[1], **Ajay Patel**[2], **Kanishk Singh**[1],
Chris Callison-Burch[2], **Kathleen McKeown**[1], **Zhou Yu**[1]
[1]Columbia University, [2]University of Pennsylvania

zfh2000@columbia.edu, ajayp@seas.upenn.edu, ks4038@columbia.edu
ccb@seas.upenn.edu, kathy@cs.columbia.edu, zy2461@columbia.edu

Abstract

The goal of text style transfer is to transform the style of texts while preserving their original meaning, often with only a few examples of the target style. Existing style transfer methods generally rely on the few-shot capabilities of large language models or on complex controllable text generation approaches that are inefficient and underperform on fluency metrics. We introduce TinyStyler, a lightweight but effective approach, which leverages a small language model (800M params) and pre-trained authorship embeddings to perform efficient, few-shot text style transfer. We evaluate on the challenging task of authorship style transfer and find TinyStyler outperforms strong approaches such as GPT-4. We also evaluate TinyStyler's ability to perform text attribute style transfer (formal ↔ informal) with automatic and human evaluations and find that the approach outperforms recent controllable text generation methods. Our model has been made publicly available here.

Figure 1: TinyStyler uses authorship embeddings from examples of the target style and conditions on these to rewrite source texts to match the target style. We replace expletives above with '∗'.

1 Introduction

Text style transfer is the task of transforming a source text to match a target style while preserving its original meaning (Jin et al., 2022; Krishna et al., 2020; Patel et al., 2022; Horvitz et al., 2024). These target styles can be defined in multiple ways, including around attributes (e.g. formality) or authorship (e.g. Barack Obama) (Jin et al., 2022). Building style transfer systems is complicated by the lack of paired data between different styles (Krishna et al., 2020). For tasks like authorship transfer, there may even be limited available data in a target style (e.g. for a non-famous author), which poses an additional challenge (Patel et al., 2022) and motivates few-shot approaches.

Several recent style transfer approaches rely on prompting large language models (LLMs) (Patel et al., 2022; Reif et al., 2022). Unlike previous style transfer approaches, these LLM-based methods can perform well on arbitrary target styles, with only few examples of a target style. Style transfer utilizing LLMs depends on in-context learning (ICL) capabilities that only reliably emerge with scale in very large models (Radford et al., 2019; Wei et al., 2022; Lu et al., 2023). The inefficiency of using these large models along with long prompts packed with in-context examples limits the practical utility of these approaches. While Suzgun et al. (2022) demonstrate that inference-time ranking can improve the style transfer performance of smaller language models, they also show that

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13376–13390

large performance gaps remain, particularly for rarer target styles.

Recent controllable text generation approaches present alternatives that rely on smaller models for fine-grained control over stylistic features (Khan et al., 2024; Horvitz et al., 2024; Mireshghallah et al., 2022), however, these methods rely on slow sampling procedures or are prone to disfluencies. Moreover, these approaches are significantly more complex and cumbersome to use than prompting LLMs.

In this paper, we introduce TINYSTYLER,[1] a simple and efficient approach to few-shot text style transfer that harnesses small language models and recent advances on authorship representations (Wegmann et al., 2022; Rivera-Soto et al., 2021) that aim to capture the writing style of an author. TINYSTYLER is trained in an unsupervised fashion over a large, diverse corpus of texts to reconstruct texts from paraphrases by conditioning on an authorship embedding of the original text. At inference time, few-shot style transfer can be performed by conditioning on the authorship embedding of a new, desired target style. Inspired by RAFT (Dong et al., 2023), we further improve model performance by sampling a large number of transferred texts, filtering these texts using automatic metrics, and fine-tuning on the resulting high-quality pairs. The resulting approach enables simple, few-shot style transfer that is both on par with state-of-the-art LLMs *and* fine-grained control through interpolation of the target style embedding (Dong et al., 2023).

In summary, our contributions are as follows:

1. We introduce TINYSTYLER, a fast, efficient, and performant approach for few-shot style transfer with authorship embeddings.

2. We evaluate our approach on both authorship and formality style transfer tasks, where we find that TINYSTYLER outperforms or is competitive with strong baselines like GPT-3.5 and GPT-4, as well as outperforms recent controllable style transfer methods. Our human evaluation provides evidence that TINYSTYLER (with only 800M parameters) offers an efficient alternative to state-of-the-art LLMs and in-context learning.

3. By conditioning on an interpolation between the source style embedding and the target style embedding, we find TINYSTYLER enables fine-grained control over trading-off style transfer accuracy for meaning preservation. As a result, the approach confers many of the benefits of alternative recent controllable generation approaches (Khan et al., 2024; Horvitz et al., 2024; Mireshghallah et al., 2022).

[1]Our code is available at https://github.com/zacharyhorvitz/TinyStyler.

2 Related Work

Unsupervised Style Transfer Reconstructing text as a framework for performing unsupervised style transfer where no parallel examples between the source and target styles exist is an established pattern used in prior work (Krishna et al., 2020; Riley et al., 2021; Jangra et al., 2022; Horvitz et al., 2024). In this paper, we build on the framework introduced by Krishna et al. (2020), which first uses paraphrasing to neutralize the style of a text and then trains a reconstruction model that learns to re-stylize it. Rather than train a model per target style, we perform style transfer to different target styles with a single model by conditioning on a representation of the target style like Riley et al. (2021). Unlike both approaches, we leverage the information captured in strong, pre-trained authorship embeddings (Wegmann et al., 2022; Rivera-Soto et al., 2021).

Few-Shot Style Transfer with LLMs Several recent approaches perform style transfer with LLMs and in-context learning (Patel et al., 2022; Reif et al., 2022). Suzgun et al. (2022) investigates reranking as a method to boost the quality of outputs generated by smaller LLMs. Other methods perform knowledge distillation from larger LLMs into smaller models (Saakyan and Muresan, 2023; Zhang et al., 2024).

Controllable Text Generation Another line of recent work has applied controllable text generation approaches to style transfer (Horvitz et al., 2024; Khan et al., 2024; Kumar et al., 2021; Mireshghallah et al., 2022; Dale et al., 2021). While text diffusion approaches like PARAGUIDE (Horvitz et al., 2024) and MCMC approaches like Mix & Match (Mireshghallah et al., 2022) afford fine-grained stylistic control, their non-autoregressive sampling procedures corresponds to longer inference times and an increased risk of disfluent outputs.

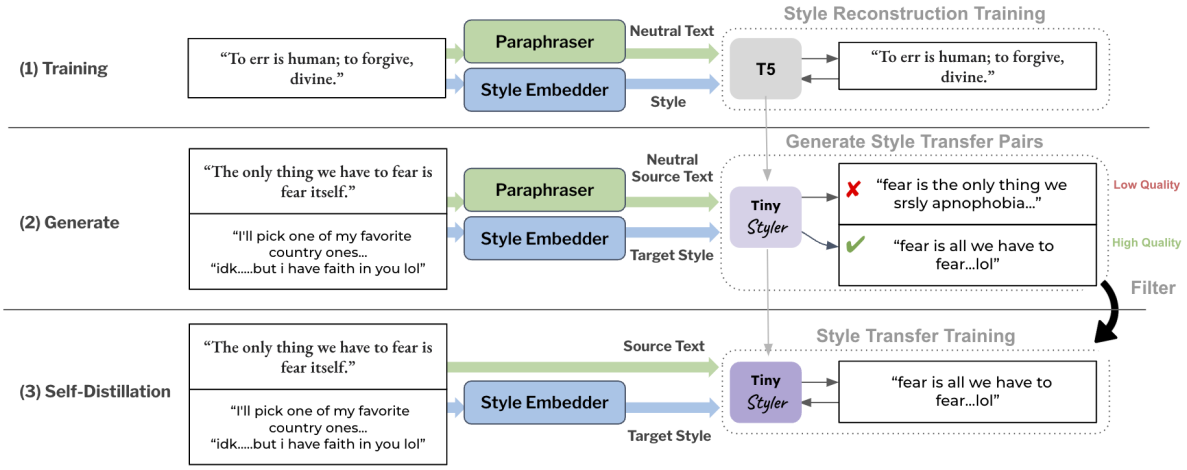

Figure 2: **Step 1)** We train a model to reconstruct texts from their paraphrases following Krishna et al. (2020), however, we only train a single model for all styles. To do this, we condition reconstruction on pre-trained authorship embeddings. **Step 2)** We generate style transfer pairs by transforming Reddit posts from a source author by conditioning generation on authorship embeddings from a different Reddit author. We filter low-quality style transfer pairs automatically using meaning preservation and stylistic similarity metrics. **Step 3)** We self-distill our model on the remaining high-quality pairs to improve the consistency of our approach and remove the reliance on a separate, external paraphrasing model.

3 TinyStyler

Our style-transfer approach is built around reconstructing texts. Following previous work (Krishna et al., 2020; Patel et al., 2022; Horvitz et al., 2024), we use paraphrasing to neutralize stylistic features from texts while preserving their original meaning. In Section 3.1, we describe training a single model to reconstruct these texts from both their paraphrases *and* their pre-trained authorship embeddings. In Section 3.2, we describe how we can then transform source texts to arbitrary target styles with the trained reconstruction model by conditioning on new authorship embeddings from texts in a target style. We use this reconstruction model to construct a dataset of high-quality synthetic style transfer pairs. Finally, in Section 3.3, we perform self-distillation to improve the consistency of our model and remove the reliance on a separate paraphrasing model. We illustrate the TinyStyler procedure in Figure 2.

3.1 Style Reconstruction Training

Following Krishna et al. (2020), we first train a model to reconstruct a text from a paraphrase of the original text with neutralized style. Instead of training a unique model per target style, we train a single reconstruction model that conditions on both a paraphrase and an authorship embedding for a source text (See Figure 2, Step 1). Later, in 3.2, we leverage this approach to build a style transfer dataset for training a stronger, simpler pipeline.

Dataset Training a general-purpose reconstruction model requires a corpus that covers many diverse authorship styles. Accordingly, we use a subset of the Reddit Million User Dataset (MUD) (Khan et al., 2021), which contains comments from over 1 million Reddit users. For each username, we sample 10 random comments and filter all comments longer than 60 tokens. The resulting dataset contains 8 million comments in total.

Authorship Embeddings We consider two approaches that learn neural representations of the writing style of authors in continuous space: 1) "Style Embeddings" (Wegmann et al., 2022) and 2) "Universal Authorship Representations (UAR)" (Rivera-Soto et al., 2021). A critical property of representations for style transfer is that they disentangle style and content. Style embeddings, for example, are trained with a contrastive authorship verification (CAV) objective, using texts from the same author on different topics as positive examples and texts on the same topic from different authors as negative examples to form training triplets. As a result, we train TinyStyler to perform style transfer by conditioning on Style embeddings. We reserve UAR embeddings, which are trained on a larger dataset, as a held-out authorship representation space for automatic evaluation of authorship style transfer following Patel et al.

(2022) and Horvitz et al. (2024).

Architecture To reconstruct texts from paraphrases and authorship embeddings, we fine-tune a modified T5 model (Raffel et al., 2020) with 800 million parameters. We adapt the model to condition on authorship style information in the authorship embedding by jointly learning a projection from the embedding's dimension ($d = 768$) to the T5 model's hidden dimension ($d = 512$). We then prepend the projected embedding to the word embeddings of the input text.

Training Details We generate paraphrases and STYLE embeddings for each comment in our corpus. To generate paraphrases, we use an off-the-shelf paraphrasing PEGASUS model (Zhang et al., 2020) with the same configuration as Horvitz et al. (2024).[2] We train the T5 model to reconstruct the original comment, conditioned on each (paraphrase, STYLE embedding) pair. We include additional details on our training procedure and hyperparameters in Appendix A.1.

3.2 Generating Style Transfer Pairs

The reconstruction model can now be used for style transfer by paraphrasing a text and using the model for reconstruction while conditioning on the STYLE embedding of a desired target style (See Figure 2, Step 2). To use multiple example texts of a target style, we combine their STYLE embeddings through a mean pool operation. This enables our approach to condition on an arbitrary number of example texts of a target style with no additional memory overhead. This initial approach to style transfer is already fast and efficient compared to text diffusion denoising (Horvitz et al., 2024) or MCMC sampling (Mireshghallah et al., 2022), and inexpensive compared with prompting LLMs. We take advantage of this reconstruction model's efficiency at performing style transfer and generate many example style transfer pairs, rerank them using automatic evaluation metrics for quality, and filter out low-quality pairs. The result is a synthetic dataset of high-quality examples of style transfer.

High Quality Dataset To build our high quality synthetic dataset, we sample 160k unique random author pairs from the Reddit MUD Dataset (Khan et al., 2021). For each pair, we choose a random source text and generate multiple outputs by sampling different paraphrases from the paraphrase model. Jangra et al. (2022) found that models trained to reconstruct from paraphrases are prone to hallucinations and we hypothesize sampling different paraphrases and reranking model outputs may mitigate these hallucinations.

Reranking and Filtering Like Suzgun et al. (2022), we rank outputs from our style transfer system using automatic metrics. To rank outputs, we utilize the automatic style transfer metrics (*Away*, *Towards*, and *Sim*) introduced by Patel et al. (2022). We rank each output per inference using the geometric mean of all three metrics ($G(G(Away, Towards), Sim)$) and select the output with the highest score. We compute *Away* and *Towards* scores using STYLE embeddings (Wegmann et al., 2022). To compute *Sim* scores, we use Mutual Implication Score (MIS), which has been shown to correlate with human judgments on style transfer tasks (Babakov et al., 2022). We then filter the resulting outputs with low scores on two meaning preservation metrics, MIS (Babakov et al., 2022) and SimCSE (Gao et al., 2022). We also filter outputs with low *Away* and *Towards* metrics, which we compute with STYLE embeddings (Wegmann et al., 2022). When multiple candidates remain, we select the highest ranked example. After filtering and selection, we are left with 40K high-quality examples of style transfer pairs. Additional details on dataset generation are included in Appendix A.2.1.

3.3 Self-Distillation on High Quality Examples

The style transfer procedure we describe in Section 3.2 with the trained reconstruction model still relies on using a paraphrase model to generate inputs for the reconstruction model. This requires performing inference with a separate model, a cumbersome procedure that also increases inference times. Additionally, while we can utilize reranking to improve performance, generating multiple candidate outputs also requires longer inference times and additional compute overhead (Suzgun et al., 2022). To address these limitations and improve the consistency of our approach, we distill away reranking and the paraphrasing step entirely by further fine-tuning our reconstruction model on the high-quality synthetic dataset generated by reconstruction model itself, essentially a self-distillation (Zhang et al., 2019) (See Figure 2, Step 3). During self-distillation, we fine-tune the reconstruction

[2]We use `tuner007/pegasus_paraphrase`.

Source	Informal	Question + Sentence	Barack Obama's Speeches
Toto, I've a feeling we're not in Kansas anymore.	*i think we arent in Kansas anymore tbh.*	*We are not in Kansas anymore? I feel you.*	*I think we are no longer in Kansas.*
is mayonnaise an instrument?	*oh wait mayonnaise is an instrument :(*	*Can you tell me what mayonnaise is? It is an instrument.*	*This makes me wonder if mayonnaise is an instrument.*
Life is like riding a bicycle. To keep your balance, you must keep moving.	*life is like riding a bicycle so you gotta keep moving :(*	*Life is like riding a bicycle. Do you have to keep moving?*	*But life is like riding a bicycle - you must keep moving.*
Life moves pretty fast. If you don't stop and look around once in a while, you could miss it.	*life moves fast if you dont stop and look around once in a while i think.*	*Is this an important lesson? Life moves fast. If you don't stop and look around once in a while, you could miss it.*	*If you don't stop and look around once in a while, you could miss it.*
The first rule of Fight Club is you do not talk about Fight Club.	*u dont talk about fight club you know,first rule.*	*First rule of Fight Club? Do not talk about Fight Club.*	*The first rule of Fight Club is not to talk about it.*

Table 1: We display TINYSTYLER outputs for various target styles. These outputs demonstrate TINYSTYLER's ability to transform text across various stylistic properties from lexical and punctuation choice to syntactic structure.

model to generate the high-quality output found through reranking and filtering and condition generation on the source text and the target author's STYLE embeddings. Additional details on our self-distillation procedure are in Appendix A.2.3.

4 Evaluation

We evaluate TINYSTYLER on authorship style transfer and text attribute style transfer (formal $\leftrightarrow$ informal).

4.1 Low-Resource Authorship Transfer

For non-famous authors, there may only be a few texts available in their authorship style. Low-resource authorship style transfer is the task of transforming to a target author's style with limited target style data. We evaluate our approach on the dataset introduced by Patel et al. (2022) using their three dataset splits (*Random*, *Single*, *Diverse*). Each split has 15 Reddit users that serve as source styles, 15 Reddit users that serve as target styles, each with 16 writing samples. In total, there are 225 (15 * 15) style transfer directions and 3600 (225 * 16) total transformations per split.

Implementation Details To perform authorship style transfer with TINYSTYLER, we condition on the target style examples using a mean pool operation over the embeddings. We consider two configurations of TINYSTYLER. $\text{TINYSTYLER}_{\text{RECON}}$ is the initial style transfer approach detailed in Section 3.2 and Figure 2, Step 2 that reconstructs from a paraphrase and target style embedding. TINYSTYLER is the final model secondarily fine-tuned on the high quality synthetic dataset described in Section 3.3 to distill away the use of the separate paraphraser model and improve consistency and quality. For each configuration, we also investigate the effect of reranking at inference time as an optional technique to further boost performance. While inference time reranking adds additional compute overhead, we note that our approach has lower inference times than other techniques (See Appendix D). Appendix B.1 discusses the authorship transfer evaluations in detail.

Metrics We evaluate on the *Away*, *Towards*, *Sim*, and *Joint* metrics from Patel et al. (2022). Notably, unlike during reranking, where we compute these same metrics like *Away* and *Towards* with STYLE embeddings, during evaluation we instead compute these metrics using the held-out UAR embeddings (Rivera-Soto et al., 2021). This avoids directly reranking on the automatic style evaluation metric, which could inflate performance.

Baselines We include results for all methods implemented by Patel et al. (2022), including LLM-based approaches like $\text{STYLL}_{\text{GPT-3}}$ and $\text{STYLL}_{\text{BLOOM}}$. Additionally, we include results for PARAGUIDE, a recent style transfer approach that

Method	Random				Single				Diverse			
	AWAY	TOWARDS	SIM	JOINT	AWAY	TOWARDS	SIM	JOINT	AWAY	TOWARDS	SIM	JOINT
COPY$_{\text{SRC}}$	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00
COPY$_{\text{TGT}}$	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.00
CAPI	0.42	0.02	0.89	0.17	0.56	0.01	0.93	0.07	0.41	0.01	0.87	0.08
CONT	0.20	0.01	0.91	0.15	0.22	0.02	0.97	0.16	0.21	0.01	0.93	0.13
SYNM	0.23	0.02	0.92	0.17	0.22	0.01	0.95	0.10	0.17	0.01	0.91	0.07
PUNC	0.23	0.02	0.93	0.24	0.25	0.02	0.97	0.19	0.26	0.02	0.90	0.18
EMOJ	0.27	0.04	0.93	0.25	0.29	0.06	0.95	0.27	0.27	0.02	0.93	0.17
PARA$_{\text{NEU}}$	0.78	0.01	0.58	0.05	0.91	0.01	0.60	0.05	0.87	0.05	0.53	0.18
PARA$_{\text{DIV}}$	0.75	0.01	0.69	0.10	0.91	0.02	0.71	0.11	0.83	0.04	0.70	0.18
LING	0.60	0.06	0.85	0.32	0.71	0.06	0.88	0.25	0.57	0.03	0.81	0.19
BERT	0.22	0.02	0.72	0.13	0.29	0.01	0.69	0.10	0.30	0.01	0.60	0.08
STRAP$_{p=0.0}$	0.98	0.02	0.16	0.05	1.00	0.00	0.23	0.00	0.97	0.02	0.22	0.04
STRAP$_{p=0.6}$	0.99	0.02	0.08	0.04	1.00	0.00	0.13	0.01	0.97	0.02	0.11	0.03
STRAP$_{p=0.9}$	0.99	0.02	0.05	0.03	1.00	0.00	0.08	0.00	0.97	0.01	0.05	0.01
PGUIDE$_{\lambda=200}$	0.70	0.05	0.58	0.22	0.82	0.06	0.66	0.26	0.77	0.06	0.54	0.22
PGUIDE$_{\lambda=800}$	0.74	0.06	0.55	0.25	0.86	0.06	0.62	0.27	0.81	0.07	0.50	0.25
PGUIDE$_{\lambda=1500}$	0.77	0.06	0.51	0.25	0.88	0.06	0.57	0.26	0.84	0.08	0.44	0.25
PGUIDE$_{\lambda=2500}$	0.80	0.07	0.46	0.25	0.90	0.06	0.51	0.23	0.86	0.08	0.38	0.24
STYLL$_{\text{GPT-3}}$	0.78	0.07	0.45	0.23	0.91	0.11	0.48	0.29	0.87	0.12	0.44	0.30
STYLL$_{\text{BLOOM}}$	0.70	0.11	0.54	0.34	0.86	0.16	0.57	0.40	0.76	0.12	0.58	0.36
GPT-3.5	0.47	0.09	0.78	0.35	0.60	0.15	0.68	0.41	0.51	0.10	0.75	0.37
GPT-4	0.76	0.09	0.71	0.33	0.85	0.08	0.72	0.31	0.83	0.07	0.68	0.30
TSTYLER$_{\text{RECON}}$	0.86	0.15	0.31	0.32	0.93	0.15	0.44	0.37	0.90	0.13	0.31	0.28
TSTYLER$_{\text{RECON,RERANK}(5)}$	0.85	0.15	0.46	0.38	0.93	0.15	0.59	0.43	0.88	0.13	0.46	0.35
TSTYLER	0.83	0.13	0.59	0.40	0.91	0.13	0.71	0.45	0.84	0.11	0.58	0.36
TSTYLER$_{\text{RERANK}(5)}$	0.84	0.12	0.70	**0.43**	0.91	0.13	0.79	**0.48**	0.83	0.11	0.69	**0.39**

Table 2: We reproduce the low-resource authorship style transfer evaluations from Patel et al. (2022) on samples from the Reddit Million User Dataset (Khan et al., 2021). We measure *Away* and *Towards* metrics using *UAR* (Rivera-Soto et al., 2021). We compute *Sim* with MIS (Babakov et al., 2022). The highest *Joint* metrics are **bolded**.

uses text diffusion models (Horvitz et al., 2024). We also include results from prompted GPT-3.5 and GPT-4. Additional details on our baseline implementations are included in Appendix B.3.

4.2 Formality Transfer

We also evaluate TINYSTYLER's ability to perform text attribute style transfer using the established GYAFC dataset (Rao and Tetreault, 2018). Authorship style transfer is difficult for humans to evaluate (Patel et al., 2022), and we select formality as an additional task because the attribute is broadly recognizable to human annotators.

Implementation Details We perform formality style transfer with TINYSTYLER by providing few-shot examples of formal or informal texts. We condition on NUM EXAMPLES of texts by extracting style embeddings for each these target style examples and then mean pooling their embeddings. We evaluate TINYSTYLER with NUM EXAMPLES $= 16$ and NUM EXAMPLES $= 64$. Several of the baselines that we compare against require a classifier trained on formal and informal texts to guide generation. We train a model for this purpose on GYAFC, and also use this classifier to select representative few-shot examples for TINYSTYLER.

Baselines We compare TINYSTYLER to PARAGUIDE (Horvitz et al., 2024) and MIX AND MATCH (Mireshghallah et al., 2022), which are two recent controllable generation approaches. Both approaches are guided at inference time by the classifier trained on GYAFC. We also benchmark against GPT-4 and GPT-3.5, as well as a naive COPY baseline for reference.

Metrics To evaluate style transfer accuracy, we use an off-the-shelf formality classifier that is held-out for evaluation (Dementieva et al., 2023; Briakou et al., 2021). Similar to our authorship style transfer evaluations, we measure meaning preservation (*Sim*) using Mutual Implication Score (MIS) (Babakov et al., 2022). Because many of the controllable approaches baselines are prone to disfluencies (Horvitz et al., 2024; Mireshghallah et al., 2022), we also compute *Fluency* scores using a model trained on the CoLA Dataset (Morris et al., 2020; Warstadt et al., 2019). We also report median *GPT-2 Perplexity* metrics which has been proposed as an alternative fluency metric (Khan et al., 2024). To compute an aggregate *Joint* metric, we follow

Method	Acc ($\rightarrow F$,$\rightarrow I$)	Sim ($\rightarrow F$,$\rightarrow I$)	Fluency ($\rightarrow F$,$\rightarrow I$)	Joint ($\rightarrow F$,$\rightarrow I$)	GPT-2
COPY$_{\text{SRC}}$	0.06 (0.10, 0.01)	0.96 (0.96, 0.97)	0.80 (0.71, 0.88)	0.05 (0.09, 0.01)	97.14
Large Language Models					
GPT-3.5	0.90 (0.97, 0.82)	0.86 (0.86, 0.87)	0.85 (0.91, 0.79)	0.79 (0.89, 0.69)	76.53
GPT-4	0.95 (0.99, 0.91)	0.89 (0.87, 0.91)	0.84 (0.91, 0.78)	0.85 (0.90, 0.80)	101.43
Controllable Text Generation Methods					
M&M$_{\text{DISC}}$	0.52 (0.12, 0.92)	0.38 (0.37, 0.38)	0.52 (0.52, 0.53)	0.24 (0.06, 0.43)	167.18
M&M$_{\text{HAM}}$	0.49 (0.08, 0.90)	0.56 (0.56, 0.57)	0.50 (0.48, 0.52)	0.29 (0.05, 0.53)	191.08
PGUIDE$_{\lambda=200}$	0.94 (0.91, 0.96)	0.65 (0.61, 0.69)	0.69 (0.68, 0.70)	0.68 (0.64, 0.71)	160.15
PGUIDE$_{\lambda=1000}$	0.97 (0.95, 0.99)	0.56 (0.47, 0.65)	0.60 (0.58, 0.63)	0.61 (0.54, 0.68)	280.54
PGUIDE$_{\lambda=5000}$	**0.97** (0.95, 0.99)	0.49 (0.37, 0.61)	0.54 (0.51, 0.57)	0.55 (0.46, 0.64)	503.46
TSTYLER	0.92 (0.88, 0.97)	0.80 (0.80, 0.81)	0.77 (0.82, 0.72)	0.76 (0.74, 0.79)	**111.58**
TSTYLER$_{\text{EX=64}}$	0.94 (0.90, 0.98)	**0.82** (0.81, 0.82)	**0.77** (0.83, 0.72)	**0.78** (0.77, 0.80)	112.5

Table 3: We evaluate formality style transfer on GYAFC and perform an automatic evaluation. We separate performance toward a formal target style ($\rightarrow F$) and informal target style ($\rightarrow I$). The best controllable approach result for each metric is **bolded**.

Method	Acc ($\rightarrow F$,$\rightarrow I$)	Sim ($\rightarrow F$,$\rightarrow I$)	Fluency ($\rightarrow F$,$\rightarrow I$)	Joint ($\rightarrow F$,$\rightarrow I$)
GPT-3.5	0.94 (0.96, 0.92)	0.97 (0.96, 0.97)	1.00 (1.00, 1.00)	0.91 (0.92, 0.89)
GPT-4	0.95 (1.00, 0.89)	0.97 (0.95, 0.99)	0.99 (1.00, 0.99)	0.91 (0.95, 0.88)
M&M$_{\text{HAM}}$	0.48 (0.05, 0.91)	0.26 (0.21, 0.31)	0.42 (0.35, 0.49)	0.11 (0.01, 0.21)
PGUIDE$_{\lambda=200}$	0.88 (0.89, 0.87)	0.48 (0.49, 0.47)	0.84 (0.85, 0.83)	0.39 (0.43, 0.36)
TSTYLER$_{\text{EX=64}}$	**0.89** (0.80, 0.97)	**0.77** (0.67, 0.87)	**0.98** (0.96, 1.00)	**0.71** (0.56, 0.85)

Table 4: Human annotator ratings of formality style transfer outputs over GYAFC on formality (*Accuracy*), meaning preservation (*Similarity*), and *Fluency*. *Joint* averages the metrics on a per-example basis.

Horvitz et al. (2024) and Krishna et al. (2020), and compute the geometric mean of *Accuracy*, *Sim*, and *Fluency*.

Human Evaluation To validate that our evaluation sufficiently aligns with human judgment of style transfer quality, we ask multiple annotators to score outputs from our approach against outputs from various approaches. We asked annotators to evaluate meaning preservation, fluency, and the formality of model outputs with binary judgements. More details describing our human evaluations are included in Appendix C.

5 Results

5.1 Authorship Transfer

Table 2 contains our authorship transfer results. Even without reranking or supervised distillation, TINYSTYLER$_{\text{RECON}}$ is competitive with the other approaches. This unsupervised approach outperforms all non-LLM baselines on *Joint* metrics. The method has strong *Away* and *Towards* metrics, but comparably low meaning preservation (*Sim*) scores. These low *Sim* scores are addressed by our refined approach, TINYSTYLER. The self-distilled TINYSTYLER achieves much higher *Joint* scores, largely due to improvements on its meaning preservation, which is comparable to LLM-based approaches. TINYSTYLER outperforms almost all baselines on *Joint* metrics. The only exception is that it slightly under-performs GPT-3.5 on the *Diverse* evaluation subset. With additional inference-time reranking, TINYSTYLER$_{\text{RERANK}(5)}$ widens this gap, and outperforms all methods on all evaluation sets.

We find TINYSTYLER demonstrates strong performance on authorship style transfer and outperforms LLMs with a notably lightweight approach, using only ~0.5% the parameters of GPT-3.5. Additionally, our experiments ablating reranking demonstrate its effectiveness in improving meaning preservation.

5.2 Formality Transfer

Tables 3 and 4 contain formality transfer results. In both automatic and human evaluations, TINYSTYLER outperforms all controllable baselines on *Joint*, *Fluency*, *GPT-2 Perplexity* and *Sim*

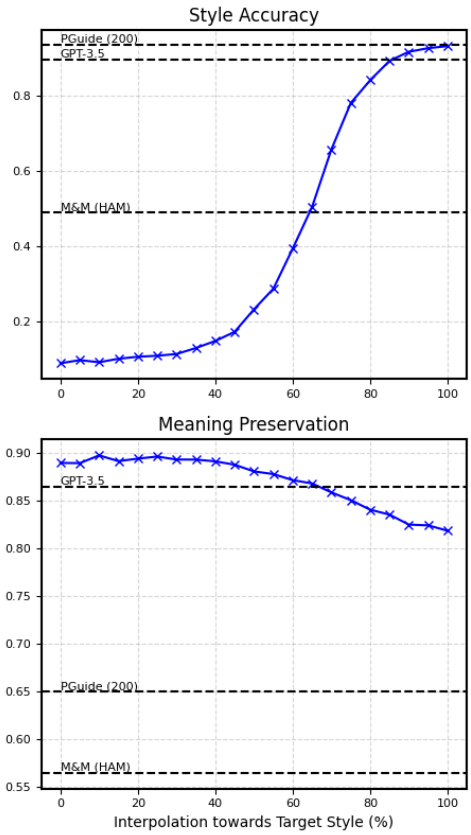

Figure 3: TINYSTYLER affords control over the strength of style transfer by interpolating between the source and target styles in STYLE embedding space. The effect on style transfer metrics for different degrees of interpolation are visualized using GYAFC.

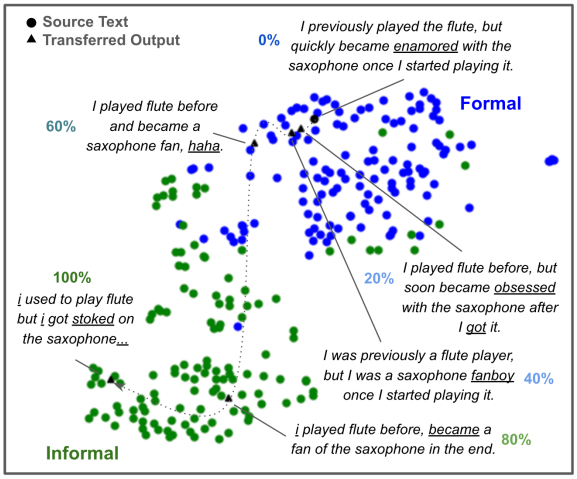

Figure 4: We transform a formal text in GYAFC by interpolating (0% to 100%) towards the average embedding of the informal texts with TINYSTYLER. We visualize the outputs alongside samples from the corpus using a t-SNE projection (van der Maaten and Hinton, 2008). Texts are embedded with STYLE embeddings.

metrics. Because TINYSTYLER aggregates target style embeddings via mean pooling, the approach can condition on additional target examples with no memory overhead. This additional conditioning in TSTYLER$_{EX=64}$ corresponds to further improvements on *Accuracy* and *Joint* metrics. While PARAGUIDE has comparable *Accuracy* scores to TINYSTYLER, the approach was rated as significantly less fluent and meaning preserving.

Considering all approaches, GPT-4 and GPT-3.5 are most performant on *Joint*, *Sim*, and *Fluency* metrics. The strong performance of LLMs on this dataset is unsurprising, as these models are trained on a large portion of the internet (OpenAI, 2023, 2022), which contains many examples of informal and formal texts. Contamination of GYAFC data or other formal re-writing examples in instruction-tuning supervised training data may also inflate performance of these models (Sainz et al., 2023). Despite these disadvantages, TINYSTYLER performs competitively with these larger models on *Accuracy* and *Fluency* ratings. Additionally, on informal transfer, TINYSTYLER performs on par with GPT-4 on the *Joint* metric (0.85 vs 0.88).

Inference Timing We include inference timing results in our Appendix. TINYSTYLER offers sub-second inference times on a single A100 GPU, and runs >35x faster than all controllable baselines. The approach is >1.5x faster than the LLM methods, which we evaluated through APIs. These results indicate that TINYSTYLER can be readily deployed in time-constrained practical applications.

5.3 Interpolating in Style Space

Like other controllable text generation approaches (Khan et al., 2024; Horvitz et al., 2024; Mireshghallah et al., 2022), TINYSTYLER enables specifying target styles at inference time. However, another advantage of these controllable methods over prompt-based style transfer is that they enable direct control of the trade-offs between style transfer and meaning preservation. In Figure 3, we visualize the effect of interpolating in STYLE embedding space style on formal ↔ informal transfer metrics. These results indicate that TINYSTYLER also affords control of the balance between metrics. Moving away from the source text embedding and towards the target style increases style transfer accuracy at the expense of meaning preservation. In Figure 4, we visualize the effect of embedding interpolation on the output text, where movement

in stylistic space corresponds to typographical, lexical, and syntactic changes.

6 Conclusion and Future Work

We introduce TINYSTYLER, a fast, simple to use, efficient approach to few-shot style transfer that uses small language models and pre-trained authorship embeddings. The method outperforms strong baselines, including GPT-4, on authorship transfer. The method also outperforms other controllable approaches on formality transfer, and is more competitive with LLMs. Our work highlights the utility of pre-trained standalone authorship embeddings and we look forward to future work on representations that capture more diverse characteristics of text style. Additionally, TINYSTYLER showcases the potential value of reranking with automatic evaluation metrics, filtering, and self-distillation as a procedure towards training efficient models that can compete with and close the performance gap with LLMs. Accordingly, we are enthusiastic about the potential for future improvements to automatic text evaluation yielding performance benefits when incorporated into text style transfer and other text generation pipelines.

7 Limitations

TINYSTYLER leverages pre-trained authorship embeddings. While TINYSTYLER can benefit from continued advances in authorship style representation learning, the approach is also bottlenecked by their current representational capacity. As a result, TINYSTYLER may underperform on more rare stylistic choices (e.g. iambic pentameter) that are not captured by STYLE embeddings. Additionally, authorship style transfer metrics may not fully capture the preferences of human authors.

8 Ethical Considerations

TINYSTYLER is an efficient approach for few-shot style transfer that consumes far fewer resources than alternative LLM-based methods. Consequently, the method can empower individuals and organizations to rewrite texts or personalize generic outputs from chat models to match a user's preferences. Simultaneously, efficient text style transfer can aid malicious actors with impersonation. Accordingly, research on text style transfer warrants a renewed focus on AI generated text detection and investment in the media literacy of the broader public.

9 Acknowledgements

We would like to thank Rahul Aditya, Amith Ananthram, Debasmita Bhattacharya, Yanda Chen, Nicholas Deas, Fei-Tzin Lee, Melanie Subbiah, Elsbeth Turcan, Haoda Wang and Yunfan Zhang for help with our human evaluations. We would also like to thank our anonymous reviewers for their constructive feedback. This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Nikolay Babakov, David Dale, Varvara Logacheva, and Alexander Panchenko. 2022. A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 300–321, Dublin, Ireland. Association for Computational Linguistics.

Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. 2021. Xformal: A benchmark for multilingual formality style transfer. *Preprint*, arXiv:2104.04108.

David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text detoxification using large pre-trained neural models. *Preprint*, arXiv:2109.08914.

Daryna Dementieva, Nikolay Babakov, and Alexander Panchenko. 2023. Detecting text formality: A study of text classification approaches. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 274–284, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *Preprint*, arXiv:2304.06767.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings. *Preprint*, arXiv:2104.08821.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *Preprint*, arXiv:1904.09751.

Zachary Horvitz, Ajay Patel, Chris Callison-Burch, Zhou Yu, and Kathleen McKeown. 2024. Paraguide: Guided diffusion paraphrasers for plug-and-play textual style transfer. *Preprint*, arXiv:2308.15459.

Anubhav Jangra, Preksha Nema, and Aravindan Raghuveer. 2022. T-STAR: Truthful style transfer using AMR graph as intermediate representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8805–8825, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Aleem Khan, Elizabeth Fleming, Noah Schofield, Marcus Bishop, and Nicholas Andrews. 2021. A deep metric learning approach to account linking. *Preprint*, arXiv:2105.07263.

Aleem Khan, Andrew Wang, Sophia Hager, and Nicholas Andrews. 2024. Learning to generate text in arbitrary writing styles. *Preprint*, arXiv:2312.17242.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *Preprint*, arXiv:2010.05700.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Preprint*, arXiv:2108.01850.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. Are emergent abilities in large language models just in-context learning? *Preprint*, arXiv:2309.01809.

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

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report.

Ajay Patel, Nicholas Andrews, and Chris Callison-Burch. 2022. Low-resource authorship style transfer with in-context learning. *Preprint*, arXiv:2212.08986.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 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)*, pages 837–848, Dublin, Ireland. Association for Computational Linguistics.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David Uthus, and Zarana Parekh. 2021. TextSETTR: Few-shot text style extraction and tunable targeted restyling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3786–3800, Online. Association for Computational Linguistics.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arkadiy Saakyan and Smaranda Muresan. 2023. Iclef: In-context learning with expert feedback for explainable style transfer. *Preprint*, arXiv:2309.08583.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2195–2222, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same author or just same topic? towards content-independent style representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Chiyu Zhang, Honglong Cai, Yuezhang, Li, Yuexin Wu, Le Hou, and Muhammad Abdul-Mageed. 2024. Distilling text style transfer with self-explanation from llms. *Preprint*, arXiv:2403.01106.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint*, arXiv:1912.08777.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. *CoRR*, abs/1905.08094.

A TinyStyler Details

A.1 Style Reconstruction Training

These sections provide details on training TinyStyler$_{\text{RECON}}$ to reconstruct texts from paraphrases and Style embeddings.

A.1.1 Dataset

We construct our training and validation datasets from the publicly available Reddit Million User Dataset (Khan et al., 2021). We reduce the size of the dataset by randomly sampling 10 comments per user. We then filter all comments longer than 60 tokens using the Pegasus tokenizer (Zhang et al., 2020), resulting in 8 million texts. We divide users into train/validation/test $(0.90, 0.05, 0.05)$, and ensure that no users from our evaluation datasets are included in the set of training authors.

A.1.2 Paraphrase and Style Embedding Generation

We generate paraphrases with a popular off-the-shelf Pegasus model (Zhang et al., 2020) that was fine-tuned for paraphrasing.[3] We use the same paraphrase model and inference hyperparameters as in ParaGuide (Horvitz et al., 2024). To paraphrase each text in our training corpus, we sample from the paraphrase model by perfoming nucleus sampling (Holtzman et al., 2020) with `top-p` $= 0.80$ and $\tau = 1.5$, on a beam search of size 8. We extract Style embeddings (Wegmann et al., 2022) for each comment in the 8 million sample using the publicly available checkpoint and inference logic.[4]

A.1.3 Architecture

We modify a T5-Large (Raffel et al., 2020)[5](800 million parameters) to condition on an input text and Style embedding. To incorporate the Style embedding, we project the vector from $d = 768$ to the model's embedding size ($d = 412$), and preprend the result to the input word embeddings.

A.1.4 Training Hyperparameters

We fine-tune the modified T5 (Raffel et al., 2020) model with the hyperparameters in Table 5 on an NVIDIA-A100 GPU. We performed minimal hyperparameter tuning, instead using established learning rates and batch sizes from previous work (Horvitz et al., 2024). This model is trained to reconstruct the original comment from its paraphrase and Style embedding. We jointly learn the Style embedding projection alongside the other model parameters.

[3] https://huggingface.co/tuner007/pegasus_paraphrase

[4] https://huggingface.co/AnnaWegmann/Style-Embedding

[5] https://huggingface.co/google/t5-v1_1-large

Hyperparameter	Value
Pretrained Ckpt	`google/t5-v1_1-large`
Learning Rate	1×10^{-5}
Batch Size	16
Grad Accum.	4
Optimizer	Adam
Weight Decay	0.01
Schedule	Constant
Warm-up Steps	2000
Total Steps	230,000

Table 5: Fine-tuning hyperparameters for TINYSTYLER$_{\text{RECON}}$.

A.2 Self-Distillation Details

The following sections describe the data generation and fine-tuning procedure for self-distillation.

A.2.1 Data Generation with TINYSTYLER$_{\text{RECON}}$

To build our **High Quality Dataset**, we follow the following procedure:

1. We first randomly sample 160k unique random (source, target) author pairs from our training dataset.

2. For each source author, we sample a single source text.

3. For each source text, we generate 5 paraphrases (As in A.1.2).

4. For each of these paraphrases, we sample $[4, 8]$ texts for a target author, extract their STYLE embeddings and mean pool the results.

5. Using these 5 pairs of (paraphrase, STYLE embedding), we sample 5 corresponding outputs from TINYSTYLER$_{\text{RECON}}$ with `top-p` $= 0.80$ and $\tau = 1.0$.

A.2.2 Filtering

After generating 5 candidate outputs for each of our 160k author pairs, we filter low quality outputs.

1. First, we filter all candidates that are identical to their input, and those with hallucinated links, which we filter with regex rules.

2. Next, we filter all candidates that have low meaning preservation scores. We use MIS (Babakov et al., 2022) and SimCSE (Gao et al., 2022), and normalize SimCSE scores between $[0, 1]$. After reviewing example outputs for errors and hallucinations, we selected 0.7 as the threshold for both models.

3. After filtering candidates with low automatic meaning preservation scores, we filter the remaining outputs with low transfer accuracy scores. We compute *Away* and *Towards* metrics using STYLE Embeddings. We filter outputs that have an *Away* < 0.9 and *Towards* < 0.30.

4. Finally, when multiple candidates remain for a given source text, we select the output that maximizes $G(G(Away, Towards), Sim)$. Here, unlike Patel et al. (2022), we skip normalizing by $MIS(source, target)$ when computing *Sim*.

The resulting data comprises approximately 40k high quality examples.

A.2.3 Fine-tuning for TINYSTYLER

We resume training our model from Appendix A.1, on the resulting High Quality Dataset. Unlike the previous train, we skip paraphrasing and condition directly on the source text, along with all texts from a target author. We otherwise use the same hyperparameters in Table 5. We select the checkpoint with the lowest validation loss, which occurred after 20000 steps.

B Evaluation Details

B.1 Authorship Transfer

We evaluate authorship transfer on the dataset introduced by Patel et al. (2022). We evaluate on their three dataset splits:

- *Random*: Random source and target authors.
- *Single*: All posts belong to a popular college football subreddit.
- *Diverse*: Source and target authors with posts on diverse topics across 13 or more different subreddits.

Each split contains 15 Reddit source authors and 15 Reddit target authors. We ensure that all authors in these evaluation sets are excluded from the training data.

When performing authorship style transfer with TINYSTYLER$_{\text{RECON,RERANK(5)}}$, we sample 5 paraphrases. Then, for each paraphrase, we sample 8 texts by the target author, and extract STYLE embeddings from these. We then generate an output for each (paraphrase, STYLE embedding) pair, and select the output with the highest value of $G(G(Away, Towards), Sim)$. We compute these metrics using STYLE embeddings (Wegmann et al., 2022), rather than UAR (Rivera-Soto et al., 2021). For TINYSTYLER$_{\text{RECON}}$, we select the first result.

For TINYSTYLER, we condition on the source text and all 16 examples for the target author. When additionally performing ranking for TINYSTYLER$_{\text{RERANK(5)}}$, we sample 5 outputs and again select the result with the highest aggregate score.

B.2 Formality Transfer

To evaluate formality transfer, we consider the Entertainment and Music subset of the GYAFC dataset (Rao and Tetreault, 2018). For all approaches, we consider the 1082 original formal examples and the 1416 original informal examples.

To measure *Accuracy*, we use a holdout off-the-shelf model,[6] trained on the XFormal Corpus (Dementieva et al., 2023; Briakou et al., 2021). For *Fluency*, we use a model[7] trained on the CoLA dataset (Morris et al., 2020; Warstadt et al., 2019).

For our internal Formality classifier, we fine-tune a `roberta-base` (Liu et al., 2019) model with the hyperparameters in Table 6 on 85% of the GYAFC Entertainment Music training set, and validate on the remaining training samples.

We use our internal classifier for our PARAGUIDE and M&M approaches (See Appendix B.3). We also use our internal classifier to determine high probability examples of each class (with probability > 0.95) to serve as exemplars for the LLM methods and TINYSTYLER models. We randomly select 128 of these formal and informal exemplars from the Tune set of GYAFC. For TINYSTYLER, along with GPT-4, and GPT-3.5, we sample 16 of these to use as examples for each inference. We sample 64 per inference for TINYSTYLER$_{\text{EX=64}}$.

[6] https://huggingface.co/s-nlp/xlmr_formality_classifier

[7] https://huggingface.co/textattack/roberta-base-CoLA

Hyperparameter	Value
Pretrained Ckpt	`roberta-base`
Learning Rate	5×10^{-5}
Batch Size	128
Optimizer	Adam
Weight Decay	0.01
Schedule	Constant
Total Steps	2700

Table 6: Fine-tuning hyperparameters for the Internal formality classifier used for MIX & MATCH and PARAGUIDE baselines.

B.3 Additional Baseline Details

For our authorship evaluations, we use the baselines from Patel et al. (2022). Additionally, we reproduce PARAGUIDE by fine-tuning the publicly available SSD-LM (Li et al., 2022) checkpoint[8] on the training dataset described in Appendix A.2.1. We extend the input and output token lengths to 80, but otherwise use the original paper's hyperparameters.

Hyperparameter	Value
Pretrained Ckpt	`xhan77/ssdlm`
Learning Rate	5×10^{-6}
Batch Size	64
Grad Accum.	2
Optimizer	Adam
Weight Decay	0.01
Schedule	Constant
Diffusion Steps	200
Context Size	80
Output Size	80
Warm-up Steps	2000
Total Steps	2000000

Table 7: Fine-tuning hyperparameters for the PARAGUIDE baseline

.

For PARAGUIDE, we perform authorship transfer using style guidance from STYLE embeddings (Wegmann et al., 2022; Horvitz et al., 2024). For MIX AND MATCH (Mireshghallah et al., 2022) we use the hyperparameters for the *Hamming* and *Disc* configurations from the original paper for the formality transfer task. Additionally, we use `RoBERTA-Large` (Liu et al., 2019) as the base lan-

[8] https://huggingface.co/xhan77/ssdlm

guage model. For both PARAGUIDE and MIX & MATCH we use our internal classifier trained on the training subset of GYAFC (Rao and Tetreault, 2018). We estimate a total compute budget of 600 GPU hours for all TINYSTYLER and baseline experiments.

Finally, we also prompt GPT-3.5 (`gpt-3.5-turbo-0125`) and GPT-4 (`gpt-4-turbo`) for both authorship and formality transfer, using the prompts included in Appendix E.

C Human Evaluation

For our human evaluation, we generate outputs for 150 examples (75 $\rightarrow$ *Informal*, 75 $\rightarrow$ *Formal*) for each approach. We recruited 11 English speakers, all of whom were graduate student volunteers. We divided the annotations among these speakers, assigning three annotators per example. We assign labels based on majority vote. Our instructions to human annotators are included in Appendix F.

We measure inter-annotator agreement with Krippendorff's α:

Label	**Krippendorff's α**
Meaning Preservation	0.55
Fluency	0.58
Formality	0.61

Table 8: Inter-annotator agreement, computed with Krippendorff's α.

D Timing

In Table 9, we report timing results on 200 examples from the Formal $\rightarrow$ Informal transfer task. To estimate timing information, we generate outputs for 200 samples. For each local approach, we perform inference on an NVIDIA-A100 GPU.

E Prompts

We prompt GPT-3.5 and GPT-4 for all tasks using OpenAI's chat completions API.[9]

[9] https://platform.openai.com/docs/guides/text-generation/chat-completions-api

Method	Seconds/Iter
Large Language Models	
GPT-3.5	0.70
GPT-4	1.56
Controllable Text Generation	
M&M$_{\text{DISC}}$	69.3
M&M$_{\text{HAM}}$	68.2
PGUIDE$_{\lambda=200}$	17.72
TSTYLER	0.47
TSTYLER$_{\text{EX}=64}$	0.43

Table 9: Timing information on a sample of the Formal $\rightarrow$ Informal task ($n = 200$).

E.1 Authorship Transfer

```
message='The following comments are
    written by a single author: \n'
for i, text in enumerate(examples):
    message +=
        json.dumps({'text':text})+'\n'
message += "\n\nCan you rewrite the
    following comment to make it look
    like the above author's style:\n"
message +=
    json.dumps({'text':original_text})+'\n'

client.chat.completions.create(
    model=model_name,
    response_format={ "type":
        "json_object" },
    messages=[
    {"role": "system", "content": "You
        are a helpful assistant designed
        to output JSON."},
    {"role": "user", "content": message}
    ]
```

E.2 Formality Transfer

```
message = f'The following texts are
    written in {target_style} style: \n'
for i, text in enumerate(examples):
    message +=
        json.dumps({'text':text})+'\n'
message += f"\n\nCan you rewrite the
    following text to make it look like
    the above {target_style} style:\n"
message +=
    json.dumps({'text':original_text})+'\n'

client.chat.completions.create(
    model=model_name,
    response_format={ "type":
        "json_object" },
```

```
    messages=[
        {"role": "system", "content":
            "You are a helpful assistant
            designed to output JSON."},
        {"role": "user", "content":
            message}
    ]
```

F Human Evaluation Instructions

Note: These examples are selected directly from GYAFC (Rao and Tetreault, 2018), which contains offensive content.

Instructions:

```
Each annotator has been assigned a
    series of very short texts to
    review. Each example consists of a
    reference and output text.
We would like you to evaluate the output
    text across three criteria:
1) Similarity to the reference. Is the
    meaning of the reference preserved
    by the output? (0=No, 1=Yes)
2) Well-formedness/Fluency. Does the
    output look like a text that could
    reasonably appear on an internet
    forum? Is it a coherent?
    (0=Badly-Formed, 1=Well-formed)
3) Formality. Is the output text
    informal or formal? (0=informal,
    1=formal)

Empty outputs can be marked with all 0s.
Please avoid consulting other
    annotators/annotations.
```

Examples of formal text:

```
I like Rhythm and Blue music.
There's nothing he needs to change.
It does not exist.
Mine is book by Steve Martin called 'The
    Pleasure of my Company'.
What differentiates a mosquitoo from a
    blonde?
They're pretty good. Also, that's a good
    song.
I do not think Beyonce can sing, dance,
    or act. You mentioned Rihanna, who
    is that?
I was unaware that you were in law
    enforcement, as well.
I called to say 'I Love You
I would most likely not vote for him,
    although I believe Melania would be
    the most attractive First Lady in
    our country's history.
```

Examples of informal text:

```
Is Any Baby Really A Freak.
aspen colorado has he best music
    festivals, you sit all over the
    moutians its on and just hang out
You can get almost anything on ebay!
everybody is Dying to get in
not idiots like 50 cent and his whole
    Gay unit.those kinds of ppl give
    hip-hop a bad name.
different from what I've seen though
I want to be on TV!
dont let anyone decide the fate but you.
50 is just riding coattails with that
    movie.
The blind klan guy is hilarious!
```