Interpreting Style Representations via Style-Eliciting Prompts

Junghwan Kim and David Jurgens
University of Michigan
{kimjhj,jurgens}@umich.edu

Abstract

Style representation learning is a powerful tool for authorship analysis and modeling writing style, yet the latent nature of learned representations makes them difficult to interpret. Recent work has attempted to explain these representations by generating natural language descriptions with large language models (LLMs) conditioned on input text. However, such descriptions are often prone to the LLM's biases and hallucinations, and they lack an explicit objective and practical utility. In this work, we propose a novel framework for interpreting style representations through style-eliciting prompts: natural language instructions designed to steer LLMs to generate text that reflects specific stylistic attributes. We curate 1,010 distinct style features spanning 26 stylistic categories and construct a dataset by prompting an LLM to generate text conditioned on these features. Using this data, we train a decoder to generate a style prompt from the style representation of the generated text. We evaluate our approach on three tasks: (1) recovering original style prompts from generated text, (2) generating text in the same style using the recovered prompts, and (3) steering LLM outputs to match the style of human-written texts. Experiments demonstrate that our method consistently outperforms strong baselines that directly prompt LLMs with target text, achieving superior performance in both style description and style imitation. These results highlight that style-eliciting prompts can provide a practical and interpretable interface to stylistic information encoded in style representations.

1 Introduction

Writing style is a core dimension of natural language, influencing how messages are interpreted, remembered, and disseminated across various contexts (Kelly et al., 2003; Boghrati et al., 2023). To model stylistic variation computationally, recent work has developed *style representations*—vector embeddings designed to encode stylistic properties (Rivera-Soto et al., 2021; Wegmann et al., 2022; Patel et al., 2025). These representations have proven effective for modeling and comparing writing styles (Neelakanteswara et al., 2024; Soto et al., 2024; Horvitz et al., 2024a,b). Nevertheless, their latent nature obscures which stylistic attributes they encode, restricting how users can interact with them for controlled text generation.

One intuitive approach to creating interpretable style representations is to ask LLMs to describe the input text in natural language (Patel et al., 2023; Alshomary et al., 2025). Yet, free-form LLM-generated descriptions often overlook important stylistic nuances and can be influenced by model-specific biases or hallucinations (Ramnath et al., 2025). More importantly, such descriptions are primarily explanatory rather than functional: it is not obvious how they can be reliably used to reproduce, manipulate, or transfer writing style.

In this work, we adopt a complementary perspective that emphasizes control as an explanation, building on LLMs' well-established capability to follow stylistic instructions when generating text (Reif et al., 2022). We introduce a framework that interprets neural style representations as human-language style prompts—explicit natural language instructions that specify stylistic constraints and can be directly followed by LLMs. Our approach constructs a supervised learning setup in which texts are first generated from known style prompts, after which a decoder is trained to recover those prompts from the style representations of the generated texts. By grounding interpretation in prompts that are directly usable for generation, this formulation yields an operational interface for stylistic control, supporting applications such as creative writing, personalized messaging, and persona simulation (Mou and Vechtomova, 2020).

To support this approach, we build a large-scale synthetic dataset comprising 1,010 distinct style

Findings of the Association for Computational Linguistics: ACL 2026, pages 41030–41048

features organized across 26 stylistic dimensions, including lexical choice, syntactic structure, tone, and rhetorical strategy. Using these prompts, we generate 1.8M stylized responses with an LLM, forming paired examples of text and explicit stylistic instructions. This dataset provides fine-grained supervision for learning to decode style representations and enables systematic evaluation of both style characterization and controllability.

We demonstrate the benefits of our framework across three tasks: (1) inferring original style prompts from LLM-generated text, (2) producing stylistically similar outputs using inferred style prompts, and (3) steering LLMs to emulate the style of non-synthetic, human-authored texts. Across all evaluations, our method consistently surpasses baselines that rely on directly prompting LLMs with target text alone. Our method achieves substantial gains in style prompt recovery (76.0% ROUGE-1, 21.7% LaBSE, and 42.8% LLM-as-judge improvements) and stronger style alignment for style control (12.9% and 26.1% L2 improvement for LLM-generated and human-written references, respectively). Together, these results demonstrate that decoding style representations into actionable prompts provides an effective pathway for both analyzing and manipulating writing style.

Our contributions are summarized as follows: (1) We introduce a decoder-based framework that translates latent style representations into natural language style prompts, enabling interpretable and controllable use of stylistic information. (2) We introduce a new, large-scale synthetic dataset of 1.8M stylized texts paired with diverse, compositional style prompts spanning 26 stylistic dimensions. (3) We show that our method substantially outperforms existing baselines on both style prompt recovery and style control tasks. (4) We release our dataset[1] and code[2] to facilitate future research on interpretable and controllable writing style modeling.

2 Related Work

Our work connects several research areas, including style representation interpretation, style description, style transfer, and prompt discovery.

Style Representation. Originally developed for authorship verification, style representation models are trained to embed texts from the same author close together while separating those from different authors (Rivera-Soto et al., 2021; Wegmann et al., 2022; Patel et al., 2025). These dense vector representations have resulted in significant performance gains for many tasks like authorship attribution over older, more interpretable methods from stylometry (cf. Rangel and Rosso, 2019; Stamatatos et al., 2022; Bevendorff et al., 2024). Although their training objective focuses on author identity, the resulting representations capture rich stylistic features beyond author-specific ones. For instance, controlled stylistic perturbations induce consistent linear shifts (Zhu and Jurgens, 2021) or clusters (Wegmann et al., 2022) in the style representation space, indicating their sensitivity to stylistic changes. Moreover, Wang et al. (2023) verifies that various non-author-specific styles in the CDS dataset (Krishna et al., 2020) can be successfully predicted from these representations.

Style Representation Interpretation. Downstream applications of style representations—such as authorship analysis and writing style modeling—often require interpretability and transparency (Tiersma and Solan, 2002; Biber and Conrad, 2019). To address this need, prior work has attempted to interpret style representations by aligning individual embedding dimensions with stylistic features (Patel et al., 2023) or by interpolating between representative examples in a latent style space (Alshomary et al., 2025). However, in both cases, the resulting style descriptions are generated by prompting LLMs directly with the input text. As a result, these descriptions may not faithfully reflect the stylistic information encoded in the style representations, nor are they tied to a concrete, reusable objective for generation or control. In contrast, our framework defines explicit ground-truth style prompts that capture the stylistic intent injected into the style representations and can be used to steer subsequent text generation, enabling effective training supervision and well-defined evaluation.

Style Description. Recent studies (Hung et al., 2023; Huang et al., 2024; Hu et al., 2024b; Ramnath et al., 2025) have explored generating style descriptions as intermediate steps for LLM-based authorship verification (Stamatatos, 2016; Tyo et al., 2022), which aims to determine whether two documents share an author. Other work (Patel et al., 2024; Yang and Carpuat, 2025) has used style descriptions to support LLM-driven style trans-

[1] https://huggingface.co/datasets/Blablablab/style-to-text

[2] https://github.com/junghwanjkim/style-decoding

fer (Jin et al., 2022; Hu et al., 2022; Mukherjee et al., 2024), where new text is generated with the content of one input and the style of another. However, as Ramnath et al. (2025) notes, such descriptions often fail to capture the full range of stylistic variation and are susceptible to biases and hallucinations inherited from the underlying LLM. Our work addresses these limitations by treating style prompts as ground-truth descriptions, yielding more grounded, goal-directed, and operational characterizations of style.

Style Transfer. The text style transfer literature has long investigated leveraging stylistic information represented as style vectors (Hu et al., 2017; Shen et al., 2017; Prabhumoye et al., 2018; Xu et al., 2020; Shen et al., 2020). Recent work (Horvitz et al., 2024a,b) successfully trains text style transfer models that rely on pretrained style representations as input. While our setting may appear similar due to its focus on stylistic control, our work differs fundamentally in both goal and formulation. Rather than paraphrasing a given text to match a target style, our objective is to explicitly describe the target style and use this description to generate new text in that style. Due to this difference, unlike in style transfer, where meaning preservation is often a central requirement, our setting does not involve preserving the original content; instead, we focus on evaluating the faithfulness of the style description and its utility for stylistic change.

Prompt Discovery. The problem of discovering prompts that elicit specific behaviors from LLMs has attracted growing interest, particularly for uncovering harmful or undesirable behaviors (Perez et al., 2022; Liu et al., 2024; Hong et al., 2024). More recent work extends this paradigm to prompt search for arbitrary, user-defined objectives (Li et al., 2025). Our study addresses a related but more fine-grained challenge: discovering prompts that induce specific writing styles—such as tone, sentence structure, or rhetorical form—which has not been explored in prior prompt discovery work. General prompt discovery methods typically aim to find prompts that cause LLMs to generate a target text, often relying on reinforcement learning to explore the prompt space. In contrast, our method leverages synthetic supervision to train a decoder efficiently in a fully supervised manner.

3 Problem Formulation

We study the interpretation of style representations by mapping them into natural-language descriptions. Specifically, we formulate style representation interpretation as the task of decoding style representations into style prompts that can be used to steer LLMs to produce text in the style implicitly described by the representation. Given a style representation of an input text, the objective is to infer a natural-language prompt that (1) induces LLM generations whose style is consistent with the stylistic intent encoded in the representation and (2) provides an interpretable description that meaningfully characterizes the style. This formulation is motivated by three considerations: (i) such prompts are easily understandable by humans, (ii) they enable principled evaluation by assessing whether generated text exhibits the target style, and (iii) they are directly usable for downstream control of LLM generation.

Formally, let $\mathtt{S}$ denote a Style Representation Model (SRM) that maps an input text x to a latent style vector $\mathbf{x}$. Our goal is to learn a decoder $\mathtt{D}$ that maps $\mathbf{x}$ to a style prompt s which both characterizes the stylistic features encoded in $\mathbf{x}$ and steers a subsequent generation y to exhibit a style similar to that of x. This objective can be expressed as

$$\arg\min_{\mathtt{D}} \quad \ell(\mathbf{x}, \mathbf{y}), \tag{1}$$

where $\mathbf{y} = \mathtt{S}(y)$ denotes the style vector of the generated text y, and ℓ is a vector distance (e.g., L2). Appendix Figure A1a illustrates this formulation.

Directly optimizing Equation 1 is infeasible due to the vast and discrete nature of the prompt space. To address this challenge, we recast the problem as supervised learning using synthetic data. Specifically, we generate synthetic pairs (x, s) consisting of a stylized text x and its corresponding style prompt s, and train the decoder $\mathtt{D}$ to map the style vector $\mathbf{x} = \mathtt{S}(x)$ to the ground-truth prompt s:

$$\arg\min_{\mathtt{D}} \quad \mathcal{L}(\tilde{s}, s), \tag{2}$$

where $\tilde{s} = \mathtt{D}(\mathbf{x})$ denotes the decoded style prompt, and $\mathcal{L}$ is the token-level cross-entropy loss. This surrogate objective is depicted in Appendix Figure A1b. If $\mathtt{D}$ perfectly recovers the ground-truth prompt s, then both input text x and the generated text y in the original formulation are produced using the same prompt s. Consequently, their style vectors $\mathbf{x}$ and $\mathbf{y}$ are closely aligned, thereby minimizing the objective in Equation 1.

To construct synthetic training pairs (x, s), one possible approach is to use LLMs to describe the style of existing texts. However, such descriptions may overlook important stylistic nuances and can be affected by model-specific biases or hallucinations. Moreover, many texts may not exhibit sufficiently salient stylistic features to support reliable description. Instead, we begin with diverse style prompts and generate stylized texts using LLMs. Because LLMs demonstrate strong instruction-following capabilities for stylistic control, these prompts reliably induce the salient stylistic features expressed in the generated texts. Furthermore, since the prompts are used directly to generate the texts, they are guaranteed to be operational and can steer the style of LLM generations.

4 Dataset Construction

We construct a large-scale synthetic dataset of stylistically diverse texts paired with ground-truth style prompts that explicitly characterize their writing style. To this end, we generate LLM outputs conditioned on a wide range of style prompts and real-world questions drawn from public Question-Answering (QA) platforms, ensuring content diversity and facilitating direct comparison with human-written answers. The dataset construction process consists of three stages: (1) generating a large inventory of concrete style features, (2) curating diverse real-world questions, and (3) generating stylized responses conditioned jointly on style prompts and questions. We describe each stage below.

Style Generation. A central component of our dataset is a diverse set of modular style features that capture a broad spectrum of stylistic variation. These features can be composed into prompts to control multiple stylistic attributes.

We generate the style feature set using GPT-4o with a hierarchical prompting strategy. Directly prompting for a flat list of features leads to substantial redundancy and uneven coverage. Instead, we first define broad stylistic categories and then populate each category with fine-grained features.

Our category design draws on prior work and is further expanded using LLM assistance. We begin with 12 seed categories derived from existing literature (Fisher et al., 2024; Patel et al., 2025; Ramnath et al., 2025) and expand this set to 26 categories using GPT-4o. These categories span a wide range of stylistic dimensions, including "Sentence Structure and Syntax," "Word and Expression Usage," "Tone and Mood," and "Readability Level." The complete list of categories is shown in Table A4.

Each category is populated with style features through LLM generation, followed by manual curation. For each category, GPT-4o is prompted to generate 40 concrete, modular, and semantically distinct style features. We then manually filter redundant or overly similar entries both within and across categories. The resulting feature set contains 1,010 curated style features spanning diverse categories. Examples of these features are shown in Figure 1, with additional details and curation prompts provided in Appendix C.1.

Question Curation. We ground stylized generation in a QA setting to enable diverse outputs under the same style prompt. This setup supports open-ended yet content-grounded responses, allowing models to vary content while adhering to a specified style. By pairing a single style prompt with different questions, LLMs can generate stylistically consistent but semantically distinct outputs.

We curate 300,000 real-world questions from 3 publicly available QA platforms: Reddit, StackExchange, and Yahoo Answers. For each platform, we select 10 topics[3] and sample 10,000, 5,000, and 15,000 questions per topic from Reddit, StackExchange, and Yahoo Answers, respectively. The resulting questions span a broad range of domains, including technical problem solving, personal advice, and opinion-based discussion. The full list of topics is provided in Table A5, with further details in Appendix C.2.

To support evaluation against human-written references, we also collect corresponding human answers for the curated questions. We first filter out questions with no human responses, then randomly select one answer per remaining question.

LLM Generation. We generate stylized responses by prompting LLMs with both a style prompt and a question. Each style prompt is constructed by randomly sampling between 1 and 10 style features from the curated feature set and concatenating them into a single instruction. To avoid conflicting stylistic constraints, we ensure that features within a prompt are drawn from distinct style categories. The sampling distribution favors prompts with fewer features, enabling the decoder to first learn individual stylistic attributes before generalizing to compositional styles.

[3]Topics correspond to subreddits for Reddit and sites for StackExchange.

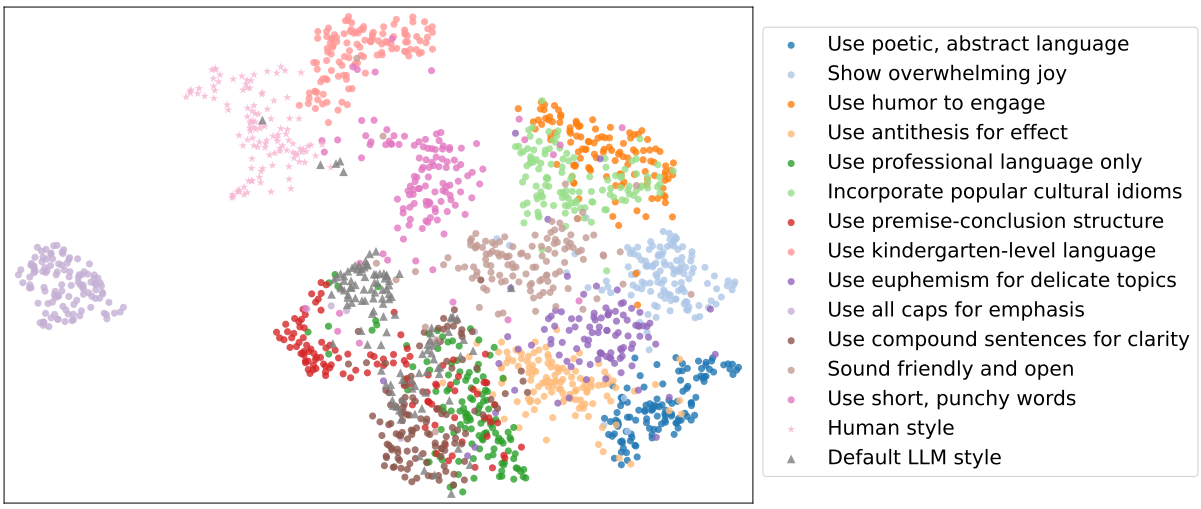


Figure 1: We visualize the style representations of a subset of our final dataset using t-SNE. Generations conditioned on different style prompts form distinct clusters, with stylistically similar clusters appearing closer together, illustrating that style representations admit style prompt recovery.

Each question is paired with two distinct style prompts, yielding two stylized responses per question. To mitigate model-specific stylistic biases, we generate responses using three different LLMs: Phi-4, Qwen2.5-14B, and OLMo-2-13B. Additional details on prompting and generation settings are provided in Appendix C.3.

Final Dataset. Each dataset entry consists of a question, a style prompt, and a stylized response. Together, these examples provide explicit supervision for training the decoder to recover style prompts from style representations. The final dataset comprises 1.8M LLM-generated responses with 434,535 unique style prompts, covering a wide range of stylistic variations and content domains. In addition, the dataset includes 300K human-written responses to support comparison with human-authored style.

Figure 1 visualizes a subset of our dataset in the style representation space. Distinct clusters emerge, each corresponding to a different style prompt, with semantically similar prompts forming nearby clusters. For instance, the prompts "Use professional language only" and "Use compound sentences for clarity" produce overlapping clusters. Notably, the default LLM style lies close to professional language, while human-written responses cluster near short, punchy wording. These patterns validate that the geometry of the style representation space captures stylistic similarity and variation, providing evidence that the representation space admits a decoding into interpretable style prompts.

5 Proposed Method

We now describe the architecture and training of our proposed style decoder. As formulated in Section 3, the goal is to learn a decoder model $\mathtt{D}$ that maps a style vector $\mathbf{x}$ to a corresponding natural-language style prompt s.

Our decoder architecture is inspired by continuous prompt tuning, which has been shown to be effective for conditioning frozen LLMs in both NLP (Li and Liang, 2021; Lester et al., 2021) and vision–language modeling (Tsimpoukelli et al., 2021). In this framework, continuous prompts encode input- or task-specific information that guides downstream generation. We extend this paradigm by encoding stylistic information captured in style vectors into continuous prompts.

The decoder $\mathtt{D}$ consists of two components: (1) a frozen LLM that interprets the stylistic signal encoded in the style vector, and (2) a trainable projection module that maps the style vector into the token embedding space of the LLM. Because the projection module is the only trainable component, this module must be equipped with sufficient capacity to express a wide range of stylistic signals. Concretely, the projection is implemented as a three-layer feedforward network with GeLU activations (Hendrycks and Gimpel, 2023), producing 20 token embeddings, a choice that we found to work well in preliminary experiments. To further contextualize decoding, we prepend a natural-language instruction that explicitly specifies the style description task alongside the projected style

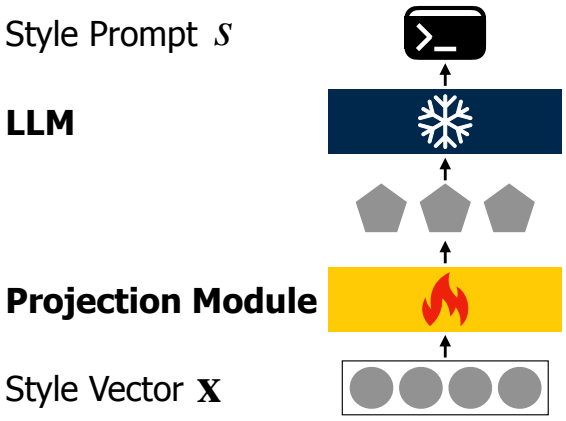


Figure 2: Our decoder D consists of a frozen LLM and a trainable projection module. The projection module maps the input style vector into the token embedding space of the LLM. Then, LLM generates a style prompt conditioned on the projected style vector.

vector, enabling the LLM to more effectively leverage its instruction-following capabilities. The instruction prompt is provided in Appendix D.1. The overall architecture is illustrated in Figure 2.

Implementation Details. For the style representation model S, we use Mistral-Nemo-Instruct-2407, trained via contrastive learning on author-labeled data following the state-of-the-art approach of Fincke and Boschee (2024). The frozen LLM used in the decoder D is Ministral-8B-Instruct.

We train the decoder D using the surrogate objective defined in Equation 2. The dataset is split into training, validation, and test sets with a ratio of 8:1:1. Training is performed for 5 epochs with a learning rate of 5e-5 and a batch size of 32. The best checkpoint is selected based on validation loss, and all reported results in this paper are obtained on the test set[4]. Additional training details are provided in Appendix D.2.

6 Recovering the Style Prompt

We first evaluate prompt recovery, the task of reconstructing the ground-truth style prompt from the style representation of a text generated by an LLM conditioned on that prompt. This task directly corresponds to the surrogate training objective defined in Equation 2. The goal of this experiment is to assess whether the proposed decoder can effectively recover the original stylistic instruction encoded in the style representation—i.e., creating an interpretable representation of the style.

6.1 Experiment Setup

Given a style prompt s and the corresponding stylized response x, our decoder maps the style representation $\mathbf{x}$ of x to the decoded style prompt $\tilde{s}$. We measure how well $\tilde{s}$ matches the ground truth s.

Baselines. We compare our approach against four baselines. The first three are LLM-based description baselines, in which an LLM is prompted to describe the writing style of a given text in natural language. Specifically, we adapt prompts from two prior works, STYLL (Patel et al., 2024) and RG (Yang and Carpuat, 2025), that generate style descriptions for style transfer, with only minor modifications necessary for our setting. Because these prompts were originally designed for style transfer, we additionally introduce a custom prompt baseline tailored to prompt recovery. To minimize performance differences arising from format mismatch, the custom baseline instructs the LLM to match the length and sentence structure of the desired output. All description baselines use Ministral-8B-Instruct, the same LLM employed in our decoder, ensuring a fair comparison. The full instruction prompts are provided in Appendix E.1. The final baseline is random sampling, where a style prompt is drawn uniformly from the same distribution used during dataset construction. This baseline serves as a naive lower bound on prompt recovery performance.

There is no straightforward off-the-shelf baseline that consumes dense style vectors. We leave the comparison of different instantiations of our framework: mapping style vector to style prompt as future work.

Metrics. We evaluate prompt recovery using three complementary metrics that capture both lexical and semantic similarity. ROUGE-1 measures unigram overlap between the decoded prompt and the ground-truth prompt. While this metric closely aligns with the training objective, it may underestimate similarity when stylistically equivalent prompts differ in surface wording. To account for this limitation, we additionally report semantic metrics. LaBSE similarity computes cosine similarity between sentence embeddings of the decoded and ground-truth prompts, while LLM-as-judge employs an external LLM (Qwen3-14B) to rate prompt similarity on a 0–10 scale. The evaluation prompt is provided in Appendix E.2. Importantly, the evaluation model is entirely disjoint from those

[4]180K LLM responses for Sections 6 and 7 and 60K human responses for Section 8

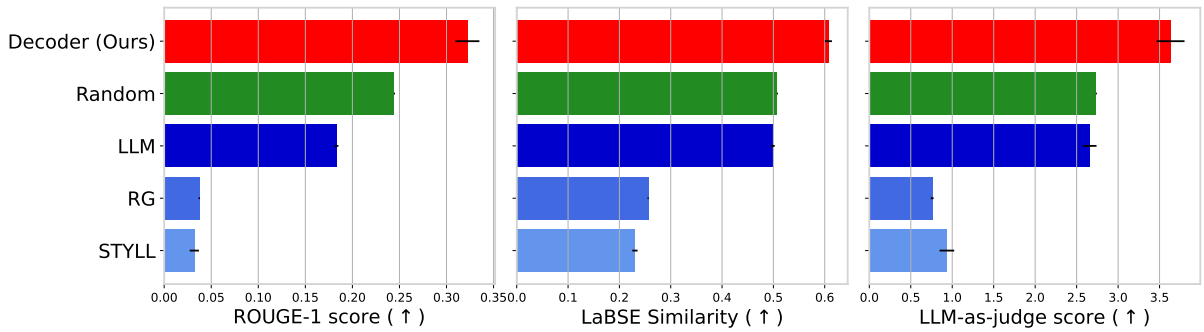


Figure 3: The style descriptions generated by our decoder match the ground truth style prompts better than the style descriptions directly generated by LLM or randomly selected style prompts in our dataset. The match is consistently demonstrated in all three metrics: ROUGE-1 score (token matching), LaBSE similarity (semantic embedding) and LLM-as-judge score (LLM judgment).

in dataset generation, decoder training, and baseline methods, ensuring an unbiased assessment.

6.2 Results

As shown in Figure 3, our decoder consistently outperforms all baselines across all three evaluation metrics by a significant margin. This result indicates that the decoder is highly effective at recovering the original style prompt in both lexical agreement and semantic similarity.

All LLM-based descriptions consistently perform worse than the random baseline. LLMs fail to generate style descriptions that capture the underlying stylistic intent expressed in the style prompt. This is consistent with the observation in Berglund et al. (2024) that even though an LLM generates B from A, it cannot infer that A could have generated B. Our custom baseline seems to benefit from output format and length information, but it still performs worse than our decoder.

Overall, these results demonstrate that decoding directly from style representations yields more accurate reconstructions of stylistic intent than post hoc LLM-based description methods. Moreover, they provide strong evidence that the style representation space encodes rich, fine-grained stylistic information, and that our decoder can effectively translate these latent signals into interpretable natural-language prompts.

Example output is shown in Table 1. From our inspection of randomly sampled examples, we do not observe content leakage—i.e., the decoded style is not somehow leaking information about the content of the text that would make inferring its style easier. Instead, all output follows the format we used to train the decoder, making our evaluative comparisons meaningful.

7 Controlling the Writing Style

We next examine whether decoded style prompts can functionally control the style of LLM outputs. In contrast to prompt reconstruction, this setting does not require recovering the original instruction verbatim. Instead, success is determined by whether the decoded prompt induces generations with the same stylistic effect as those produced under the ground-truth prompt. It therefore serves as a test of whether training with the surrogate objective in Equation 2 yields prompts that generalize to effective style control in downstream generation.

7.1 Experiment Setup

Given a style vector $\mathbf{x}$ and the style prompt $\tilde{s}$ decoded from $\mathbf{x}$, we generate a new text y conditioned on $\tilde{s}$. We measure how close the style representation $\mathbf{y}$ of y is to the original $\mathbf{x}$. To avoid information leakage, x and y are conditioned on different question contexts.

Baselines. We compare our approach against five baselines: four LLM-based style-imitation methods and one explicit style-transfer approach. For the LLM-based imitation baselines, we adapt prompting strategies from three prior works (Bhandarkar et al., 2024; Wang et al., 2025; Jangra et al., 2025), which are designed to generate responses that mimic the writing style of a given target text. We modify these prompts minimally to fit our QA setup. In addition, we again introduce a custom LLM baseline tailored to our setting. The full prompts are provided in Appendix E.1. Both our method and the LLM-based imitation baselines use Ministral-8B-Instruct for generation. As a representative style transfer baseline, we include TinyStyler (Horvitz et al., 2024b), a state-of-the-art

Example Outputs
1. The author uses unreliable memory style, uses formal vocabulary, uses interjections for surprise, uses high noun density, uses repetition for focus, uses commas to set off interrupters, uses a flowchart-like logic.
2. The author uses poetic descriptive style, uses anaphora in sentence beginnings.
3. The author uses comparative adjectives for comparison, uses minimal supporting explanation, uses antithesis in sentence structure.

Table 1: The example outputs of our decoder.

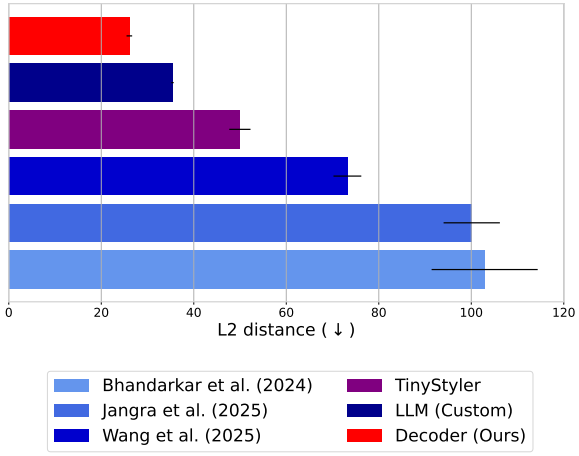


Figure 4: Our approach yields the smallest L2 distance in style representation space from the ground-truth stylized output, compared to baseline methods.

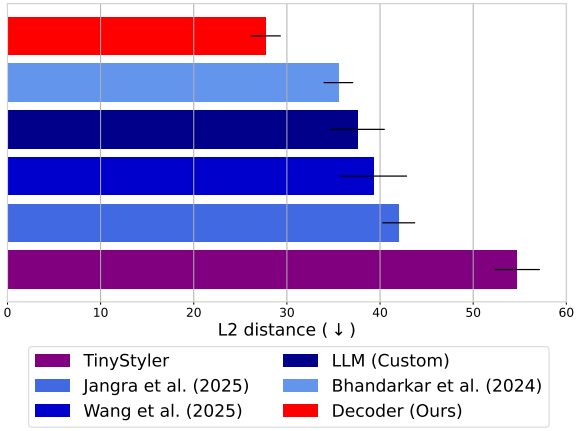


Figure 5: Our decoder outperforms baselines at generating style prompts that induce writing most similar to human-written text.

method that applies style representations to transform text. We use the official implementation, which uses its own style representation. We first generate neutral responses without style conditioning and then apply TinyStyler to transfer them into the target style.

Metrics. We assess style control by computing the L2 distance between the style representations of the original and the regenerated text. Our evaluation uses the same style representation model that encodes inputs for the decoder, since the goal of our framework is to faithfully interpret the style representation space. The decoded prompts should have a consistent effect on the target style representation space, rather than on obscure general style.

7.2 Results

Figure 4 shows that our decoder outperforms all baselines by a significant margin. This result indicates that the inferred prompts successfully translate latent stylistic information into actionable instructions that guide generation behavior.

The TinyStyler outperforms three adapted LLM-based style-imitation baselines, but underperforms our custom baseline. One plausible limitation of the TinyStyler is architectural: TinyStyler relies on a smaller T5-based generator, whereas both our decoder and the LLM-based baselines use LLMs with billions of parameters. In addition, TinyStyler employs the StyleEmbedding model (Wegmann et al., 2022), while our decoder leverages a substantially larger style representation model trained on a much broader, author-labeled corpus.

As an additional robustness test, we repeat the evaluation using two alternative style representations not used in our method, LUAR (Rivera-Soto et al., 2021) and StyleDistance (Patel et al., 2025). The results, provided in Appendix F.2, show consistent trends across all three style representations, with our approach able to generate new documents that are closer in the vector space, regardless of which representation is used. Together, these results demonstrating that the strong steering performance of our decoder is not specific to the style representation space used for training it.

8 Steering towards Human Style

Finally, we investigate whether the proposed decoder can steer LLM generation toward the writing style of human-authored text. This experiment evaluates generalization to natural, non-synthetic language outside the decoder's training distribution, which constitutes a more realistic and challenging test of practical style control.

8.1 Experiment Setup

We are given a style vector $\mathbf{x}$ and the style prompt $\tilde{s}$ decoded from $\mathbf{x}$, as in Section 7. However, the target text x is now human-authored text rather than LLM-generated text, thereby falling outside the distribution of the synthetic training data. Therefore, the decoder operates on the input distribution unseen during training, and there is no ground-truth style s that guarantees the generation in the style of x. This setting makes the task more challenging, but generalization to it is essential for the decoder to work in a more practical style control. We use the same metric and baseline as in Section 7.

8.2 Results

As shown in Figure 5, generations guided by decoded style prompts achieve substantially smaller distances to human-authored references than those produced by any baseline method. Because no ground-truth style prompts exist for human-authored texts, strong performance in this setting provides a challenging test of the proposed framework. The results indicate that the decoder can effectively interpret input style vectors and map them to appropriate style prompts even when the inputs lie outside the distribution of the synthetic training data. Moreover, the diversity of the curated style features enables generalization across a broad spectrum of stylistic variation, extending to naturally occurring human writing. Taken together, the results offer compelling evidence for the robustness and transferability of the decoding framework.

We repeat our robustness tests by using two alternative style representations LUAR and StyleDistance, to measure whether the generated style-steered document is similar to the reference human-authored document's style. Results shown in Appendix G demonstrate a consistent trend with our approach performing best for matching the style. Together, these results show that the style instructions from our decoder successfully steer style towards human texts.

9 Conclusion

We propose a novel framework for interpreting dense style representation vectors by decoding them into natural language descriptions of writing style. To train a decoder in a supervised setting, our approach leverages synthetic data generated via LLM with human-readable, compositional style prompts. Comprehensive evaluations demonstrate that the decoded prompts faithfully reflect the stylistic information encoded in style representations and can effectively guide LLM outputs to match a desired style. Our method consistently outperforms baselines across multiple metrics, including lexical and semantic match in style prompt recovery and style similarity in style control.

This work bridges the gap between latent neural representations and interpretable, controllable style attributes, offering a path toward more transparent and steerable NLP systems. Future directions include extending the method to zero-shot or unsupervised settings, improving disentanglement between content and style, and applying the decoding framework to other latent attributes such as tone, persona, or rhetorical strategy.

Limitations

Our study focuses on the English language. Since LLMs and style representation models that our framework builds on are less available or less performant in non-English languages, our method may not generalize to such settings. Moreover, writing styles in different languages can have different distributions and interactions. We hope that our work inspires other researchers to generalize our framework to non-English languages.

Our dataset is restricted to the online question-answering domain. While our decoder demonstrates strong performance in describing writing style within this context, it remains unclear how well it generalizes to other domains such as narrative writing, formal prose, or technical documentation. Evaluating the method across a broader range of domains is an important direction for future work.

Additionally, our dataset is generated using three similarly-sized LLMs. Although we observe consistent style description quality across these models, this setup does not fully capture the diversity of existing LLMs in terms of scale, architecture, and training data. To mitigate this, we test our decoder on human-written text and observe promising re-

sults. Nevertheless, caution is warranted when applying the method to text produced by models that differ significantly from those used during dataset construction, as performance may degrade in unseen generative regimes.

Ethical Considerations

This work presents a framework for analyzing the writing style of a given text and inferring a style prompt capable of eliciting similar stylistic characteristics from an LLM. While this technique can support beneficial applications—such as improved interpretability, stylistic control, and enhanced transparency in LLM-generated text—it also introduces potential risks. For example, it could be misused to undermine author anonymity in sensitive contexts or to reverse-engineer proprietary or private prompt formulations.

We emphasize that our approach is designed for controlled settings with synthetic supervision and does not aim to de-anonymize authors or recover confidential prompts. Nevertheless, we encourage future work to explore safeguards and use restrictions to mitigate misuse, particularly in high-stakes or privacy-sensitive applications.

Acknowledgments

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 #2022-22072200006. 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

Milad Alshomary, Narutatsu Ri, Marianna Apidianaki, Ajay Patel, Smaranda Muresan, and Kathleen McKeown. 2025. Latent space interpretation for stylistic analysis and explainable authorship attribution. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1124–1135, Abu Dhabi, UAE. Association for Computational Linguistics.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*.

Janek Bevendorff, Xavier Bonet Casals, Berta Chulvi, Daryna Dementieva, Ashaf Elnagar, Dayne Freitag, Maik Fröbe, Damir Korenčić, Maximilian Mayerl, Animesh Mukherjee, and 1 others. 2024. Overview of pan 2024: multi-author writing style analysis, multilingual text detoxification, oppositional thinking analysis, and generative ai authorship verification. In *European Conference on Information Retrieval*, pages 3–10. Springer.

Avanti Bhandarkar, Ronald Wilson, Anushka Swarup, and Damon Woodard. 2024. Emulating author style: A feasibility study of prompt-enabled text stylization with off-the-shelf LLMs. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 76–82, St. Julians, Malta. Association for Computational Linguistics.

Douglas Biber and Susan Conrad. 2019. *Register, genre, and style*. Cambridge University Press.

Reihane Boghrati, Jonah Berger, and Grant Packard. 2023. Style, content, and the success of ideas. *Journal of Consumer Psychology*, 33(4):688–700.

Steven Fincke and Elizabeth Boschee. 2024. Separating style from substance: Enhancing cross-genre authorship attribution through data selection and presentation. *Preprint*, arXiv:2408.05192.

Jillian Fisher, Skyler Hallinan, Ximing Lu, Mitchell L Gordon, Zaid Harchaoui, and Yejin Choi. 2024. StyleRemix: Interpretable authorship obfuscation via distillation and perturbation of style elements. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4172–4206, Miami, Florida, USA. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus). *Preprint*, arXiv:1606.08415.

Zhang-Wei Hong, Idan Shenfeld, Tsun-Hsuan Wang, Yung-Sung Chuang, Aldo Pareja, James R. Glass, Akash Srivastava, and Pulkit Agrawal. 2024. Curiosity-driven red-teaming for large language models. In *The Twelfth International Conference on Learning Representations*.

Zachary Horvitz, Ajay Patel, Chris Callison-Burch, Zhou Yu, and Kathleen McKeown. 2024a. Paraguide: guided diffusion paraphrasers for plug-and-play textual style transfer. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Zachary Horvitz, Ajay Patel, Kanishk Singh, Chris Callison-Burch, Kathleen McKeown, and Zhou Yu.

2024b. TinyStyler: Efficient few-shot text style transfer with authorship embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13376–13390, Miami, Florida, USA. Association for Computational Linguistics.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, and 5 others. 2024a. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*.

Yujia Hu, Zhiqiang Hu, Chun Wei Seah, and Roy Ka-Wei Lee. 2024b. InstructAV: Instruction fine-tuning large language models for authorship verification. In *First Conference on Language Modeling*.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. *SIGKDD Explor. Newsl.*, 24(1):14–45.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Baixiang Huang, Canyu Chen, and Kai Shu. 2024. Can large language models identify authorship? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 445–460, Miami, Florida, USA. Association for Computational Linguistics.

Chia-Yu Hung, Zhiqiang Hu, Yujia Hu, and Roy Lee. 2023. Who wrote it and why? prompting large-language models for authorship verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14078–14084, Singapore. Association for Computational Linguistics.

Anubhav Jangra, Bahareh Sarrafzadeh, Silviu Cucerzan, Adrian de Wynter, and Sujay Kumar Jauhar. 2025. Evaluating style-personalized text generation: Challenges and directions. *Preprint*, arXiv:2508.06374.

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.

Jean Kelly, Jan Knight, Lee Anne Peck, and Guy Reel. 2003. Straight/narrative? writing style changes readers' perceptions of story quality. *Newspaper Research Journal*, 24(4):118–122.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li, Neil Chowdhury, Daniel D. Johnson, Tatsunori Hashimoto, Percy Liang, Sarah Schwettmann, and Jacob Steinhardt. 2025. Eliciting language model behaviors with investigator agents. In *Forty-second International Conference on Machine Learning*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lili Mou and Olga Vechtomova. 2020. Stylized text generation: Approaches and applications. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 19–22, Online. Association for Computational Linguistics.

Sourabrata Mukherjee, Mateusz Lango, Zdenek Kasner, and Ondrej Dušek. 2024. A survey of text style transfer: Applications and ethical implications. *Preprint*, arXiv:2407.16737.

Abhiman Neelakanteswara, Shreyas Chaudhari, and Hamed Zamani. 2024. RAGs to style: Personalizing LLMs with style embeddings. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 119–123, St. Julians, Malta. Association for Computational Linguistics.

Ajay Patel, Nicholas Andrews, and Chris Callison-Burch. 2024. Low-resource authorship style transfer: Can non-famous authors be imitated? *Preprint*, arXiv:2212.08986.

Ajay Patel, Delip Rao, Ansh Kothary, Kathleen McKeown, and Chris Callison-Burch. 2023. Learning interpretable style embeddings via prompting LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15270–15290, Singapore. Association for Computational Linguistics.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. StyleDistance: Stronger content-independent style embeddings with synthetic parallel examples. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8662–8685, Albuquerque, New Mexico. Association for Computational Linguistics.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Sahana Ramnath, Kartik Pandey, Elizabeth Boschee, and Xiang Ren. 2025. CAVE: Controllable authorship verification explanations. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8939–8961, Albuquerque, New Mexico. Association for Computational Linguistics.

Felipe Rangel and Paolo Rosso. 2019. Overview of the 2019 author profiling task at pan. In *CLEF (Working Notes)*.

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.

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.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tianxiao Shen, Jonas Mueller, Dr.Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8719–8729. PMLR.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

Efstathios Stamatatos. 2016. Authorship verification: A review of recent advances. *Research in Computing Science*, 123(1):9–25.

Efstathios Stamatatos, Mike Kestemont, Krzysztof Kredens, Piotr Pezik, Annina Heini, Janek Bevendorff, Benno Stein, and Martin Potthast. 2022. Overview of the authorship verification task at pan 2022. In *CEUR workshop proceedings*, volume 3180, pages 2301–2313. CEUR-WS. org.

Peter Tiersma and Lawrence M. Solan. 2002. The linguist on the witness stand: Forensic linguistics in american courts. *Language*, 78(2):221–239.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems*, volume 34, pages 200–212. Curran Associates, Inc.

Jacob Tyo, Bhuwan Dhingra, and Zachary C Lipton. 2022. On the state of the art in authorship attribution and authorship verification. *arXiv preprint arXiv:2209.06869*.

Andrew Wang, Cristina Aggazzotti, Rebecca Kotula, Rafael Rivera Soto, Marcus Bishop, and Nicholas Andrews. 2023. Can authorship representation learning capture stylistic features? *Transactions of the Association for Computational Linguistics*, 11:1416–1431.

Zhengxiang Wang, Nafis Irtiza Tripto, Solha Park, Zhenzhen Li, and Jiawei Zhou. 2025. Catch me if you can? not yet: LLMs still struggle to imitate the implicit writing styles of everyday authors. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10040–10055, Suzhou, China. Association for Computational Linguistics.

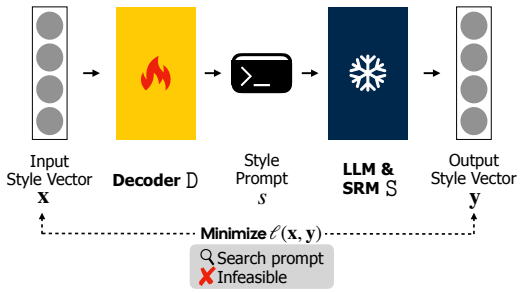
Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same author or just same topic? towards content-independent style representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.

Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision. In *Proceedings of the 37th International Conference on Machine*

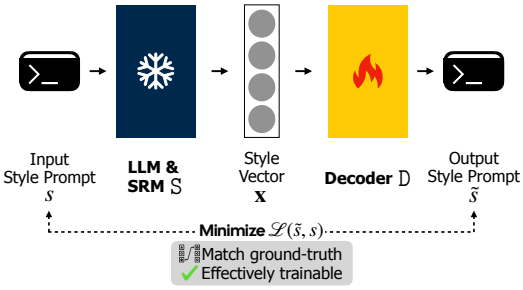
Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 10534–10543. PMLR.

Xinchen Yang and Marine Carpuat. 2025. Steering large language models with register analysis for arbitrary style transfer. *Preprint*, arXiv:2505.00679.

Jian Zhu and David Jurgens. 2021. Idiosyncratic but not arbitrary: Learning idiolects in online registers reveals distinctive yet consistent individual styles. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 279–297, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.



(a) Prompt Search Formulation in Equation 1.



(b) Prompt Recovery Formulation in Equation 2.

Figure A1: Our goal is to train a decoder D that recovers a style prompt s capable of steering an LLM to generate text y in the style of x (Figure A1a). Because searching the vast prompt space is intractable, we reformulate the task as supervised prompt recovery: Given a style-response pair (s, x), we match the decoded prompt $\tilde{s} = \mathtt{D}(\mathbf{x})$ to s (Figure A1b).

A Artifact for Reproducibility

Table A3 shows the models and their size used in our experiments.

B Supplementary Material for Section 3

Figure A1 illustrates the core problem being addressed. Instead of the prompt search formulation in Figure A1a which is infeasible, we reformulate the task as the supervised prompt recovery (Figure A1b).

C Supplementary Material for Section 4

This section provides additional details on the dataset construction process.

C.1 Style Generation

We describe here the procedure used to obtain the style features. First, we employ the **Category expansion prompt** below to obtain the style categories. Table A4 lists the stylistic categories in our dataset. Next, we apply the **Feature list prompt** below, in which `[Style Category]` is replaced with the corresponding category name to produce fine-grained style features.

Category expansion prompt.

```
Extend the list of categories for writing styles. Each category should be distinct and orthogonal from the others. Provide only new categories.

"Writing Goal and Intent", "Tone and Mood", "Grade Level", "Sentence Structure and Syntax", "Function Word Usage", "Word and Expression Choice", "Punctuation Usage", "Special Character and Capitalization Usage", "Acronym and Abbreviation Usage", "Formality", "Social and Interpersonal", "Emotional Intensity", "Descriptive Density"
```

Feature list prompt.

```
Your task is to generate a list of 40 writing styles in the [Style Category] category. Answer in JSON format, where each item is a style.

Detailed Instructions: Each style in the list should be (1) distinct and orthogonal to the others. Similar styles with subtle nuance differences are considered distinct and, therefore, allowed. (2) independent of the content of the text. (3) concrete so that LLMs can implement it when instructed to. (4) minimal so that it cannot be decomposed into multiple styles. (5) a short sentence (3-5 words) in the imperative form that instructs writers to implement the style.
```

C.2 Question Curation

To ensure diversity across content domains, we sample questions stratified by topic. For each platform, we select the 10 topics with the largest number of available questions. Table A5 presents the complete list of topics per platform.

The number of questions sampled per topic is determined proportionally to the total volume of questions available within that platform. Within each topic, both questions and corresponding human answers are sampled uniformly at random. On average, human answers in the test set contain 493.17 characters and 79.80 words.

C.3 LLM Generation

Stylized responses are generated using the **Stylized text generation prompt** below, where `[max_words]`, `[title]`, `[body]`, and `[style]` are replaced with the maximum word count, question title, question body, and style prompt, respectively.

To control generation prompt length, we truncate question titles to 80 tokens and question bodies to 200 tokens. We format all inputs using the chat template specific to the LLM used for generation. For decoding, we set the temperature to 1.0, nucleus sampling threshold (p) to 0.95, and maximum token count to 200. The maximum word count in the prompt is matched with this token count.

Stylized text generation prompt.

```
{
"system": "Your task is to answer a question following the provided style instructions. Your response should clearly show the instructed writing style, but should not directly mention any information about the style instruction. Do not explain your answer. Your answer should be within [max_words] words.",
"user": "Answer the following question while adhering to the provided writing style instructions.\n\n# Question:\n[title][body]\n\n# Style Instructions:\n[style]"
}
```

D Supplementary Material for Section 5

This section provides additional details on our proposed method.

D.1 Prompt for Decoding

We employ the **Style decoding prompt** below, where `[max_words]` is replaced with the maximum word count of 100. We concatenate `prefix`, `projected style representation`, and `suffix`, which is then provided into `V` as an input.

Style decoding prompt.

```
{
"prefix": "Your task is to describe the writing style from the style embedding vectors, in a single sentence, following this structure: 'The author [verb] [specific technique/trait], ...' You can include 1 to 10 techniques/traits, but
```

```
should not repeat. Do not explain your answer. Your answer should be within [max_words] words.\n\n### Style Embedding Vectors: \n",
"suffix": "\n\n# Description: \n"
}
```

D.2 Training Details

All models are implemented with PyTorch-Lightning and the Huggingface Transformer library. We use the AdamW optimizer (Loshchilov and Hutter, 2019) and the WSD learning rate schedule (Hu et al., 2024a). Our model training takes 16 hours using 2 A100 GPUs.

E Supplementary Material for Section 6

We provide prompts that our LLM baselines and LLM-as-judge evaluations use.

E.1 LLM baseline prompts

We show three prompts used for three LLM-based description baselines: STYLL (Patel et al., 2024), RG (Yang and Carpuat, 2025), and the custom prompt we designed. For RG (Yang and Carpuat, 2025), style descriptions were generated in two stages; thus, we present two prompts. We replace `[max_words]` and `[text]` with the maximum word count and target text, respectively. The maximum word count is set as 100, matching that of our decoder.

STYLL.

```
{
"system": "You are a forensic linguist who knows how to analyze linguistic and stylometric similarites between texts.",
"user": "List some adjectives, comma-separated, that describe the writing style of the author of this passage. Strictly output only the style descriptors without any other content.\n\n# Passage: \n[text]"
}
```

RG.

```
{
"system": "You are a forensic linguist who knows how to analyze linguistic and stylometric similarites between texts.",
"user": "Analyze the authorship style of this passage in terms of dimensions of register variation according to Douglas Biber.\n\n# Passage: \n[text]"
}
```

```
{
"system": "You are a forensic linguist who knows how to analyze linguistic and stylometric similarites between texts.",
"user": "List some adjectives, comma-separated, that describe the writing style of the author of the target text. Strictly output only the style descriptors without any other content.\n\n# Style analysis: \n[text]"
}
```

Custom Prompt.

```
{
"system": "Your task is to analyze the writing style of the text and describe the style in a single sentence, following this structure: 'The author [verb] [specific technique/trait], ...' You can include 1 to 10 techniques/traits, but should not repeat. Do not explain your answer. Your answer should be within [max_words] words.",
"user": "Describe the writing style of the text.\n\n# Text: \n[text]"
}
```

E.2 LLM-as-judge evaluation prompt

We instruct the evaluating LLM with the **LLM-as-judge score evaluation prompt** below to compute the LLM-as-judge score. We replace `[pred]` and `[ref]` with the predicted and reference styles, respectively.

LLM-as-judge score evaluation prompt.

```
You are an expert judge for text similarity. Given the following two passages, rate their semantic similarity on a scale from 0 (completely unrelated) to 10 (nearly identical in meaning).
Do not explain your answer. Provide the numeric score only.

Text A: '[pred]'
Text B: '[ref]'

Similarity score (0-10):
```

F Supplementary Material for Section 7

We provide prompts that our baselines use and additional evaluation using different style representation models.

F.1 LLM baseline prompts

We show four prompts used for four LLM-based style imitation baselines: Bhandarkar et al. (2024), Wang et al. (2025), Jangra et al. (2025), and the custom prompt we designed. We replace [max_words], [title], [body], and [reference] with the maximum word count, question title, question body, and the target text, respectively. The maximum word count is set as 200, matching that of stylized text generation.

Bhandarkar et al. (2024).

```
{
"system": "You are an emulator designed to replicate the writing style of a human author.",
"user": "# Task\nYour task is to answer the following question while seamlessly integrating with the provided human-authored snippet. Strive to make the answer stylistically indistinguishable from the human-authored text.\n\n# Instrucions\nThe goal of this task is to mimic the author's writing style while paying meticulous attention to lexical richness and diversity, sentence structure, punctuation style, special character style, expressions and idioms, overall tone, emotion and mood, or any other relevant aspect of writing style established by the author. Your answer should be within [max_words] words.\n\n# Output Indicator\nAs output, exclusively return the text completion without any accompanying explanations or comments.\n\n# Question:\n[title][body]\n\n# Human-authored Text:\n[reference]"
}
```

Wang et al. (2025).

```
{
"system": "You will be given one or more writing samples from a specific author. Your task is to analyze the author's style, tone, and voice, then craft an answer that closely mimics their writing based on a provided summary. Your writing should be around [max_words] words.",
"user": "### Author's Writing Sample\n\n[reference]\n\n### Question:\n[title][body]\n\n### Instructions\n\n- Ensure your writing faithfully replicates the author's style, including tone, word choices, and sentence structure, etc. - Maintain consistency with the author's voice while accurately reflecting the details of the given summary. - Strive to make your writing indistinguishable from the original author's work. - Do not output anything other than the writing."
}
```

Jangra et al. (2025).

```
{
"system": "You are a writing assistant. Your goal is to address to a user's query to write a text instance based on their preferences.\nThe input would comprise of the following elements enclosed in |begin INPUT|...|end INPUT| \n - |begin USER_QUERY|...|end USER_QUERY| - the user query containing the writing task description and instructions on how to generate the OUTPUT.\n - |begin STYLE_EXAMPLES|...|end STYLE_EXAMPLES| - contains the written examples that should be used for writing style, tone and voice inspiration.\n - Generate the response in |begin OUTPUT|...|end OUTPUT|.\n - Depending on the information in input generate the response accordingly.\n - Write the response based on instructions in the USER_QUERY while taking stylistic inspirations from STYLE_EXAMPLES. Responding with generic response when STYLE_EXAMPLES are present is undesirable, and therefore you should try your best to incorporate the stylistic features while not leaking any information from STYLE_EXAMPLES into OUTPUT.",
"user": "|begin INPUT|\n |begin STYLE_EXAMPLES|[reference]\n |end STYLE_EXAMPLES|\n |begin USER_QUERY| Answer the following question while mimicking the writing style of the provided reference text. Make sure
```

Method	Our Embedding	LUAR	StyleDistance
Decoder (Ours)	26.07	6.01	6.82
LLM (Custom)	35.39	9.10	8.24
Wang et al. (2025)	73.21	8.26	8.41
Jangra et al. (2025)	100.10	8.90	9.85
Bhandarkar et al. (2024)	102.89	9.02	11.87
TinyStyler	49.97	11.40	10.82

Table A1: The numeric values for evaluation in Section 7.

```
to not generate infactual information that is not present in the INPUT like dates, names, etc., and instead generate placeholders like [DATE], [NAME], etc. Your answer should be within [max_words] words.\n\n# Question:\n[title][body]\n|end USER_QUERY|\n|end INPUT|"
}
```

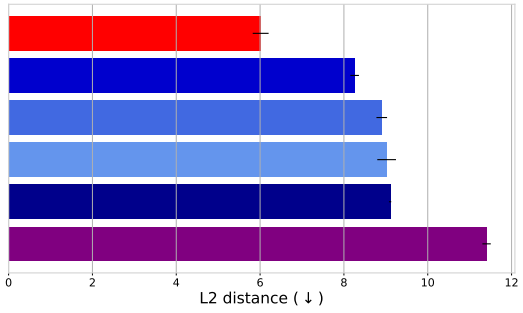
Custom Prompt.

```
{
"system": "Your task is to answer a question mimicking the writing style of the reference text. Your response should clearly show the writing style that matches that of the reference text. Do not explain your answer. Your answer should be within [max_words] words.",
"user": "Answer the following question while mimicking the writing style of the provided reference text.\n\n# Question:\n[title][body]\n\n# Reference Text:\n[reference]"
}
```

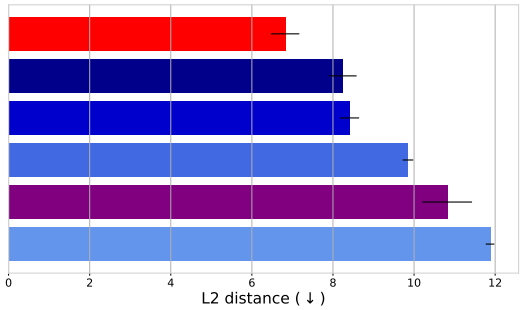
F.2 Evaluation with other style representations

As additional robustness tests, we reproduce our experiments using two alternative style representation models. Figure A2 shows the evaluation in Figure 4 with LUAR (Rivera-Soto et al., 2021) and StyleDistance (Patel et al., 2025) style embeddings for documents. Table A1 presents all numeric values for Figures 4 and A2.

The trends across different representation models agree. Since our dataset construction and decoder never use these two additional style representation models, there is no potential circularity from these representations. These results demonstrate that the strong performance of our decoder is not limited to the style representation used to train it.



(a) LUAR



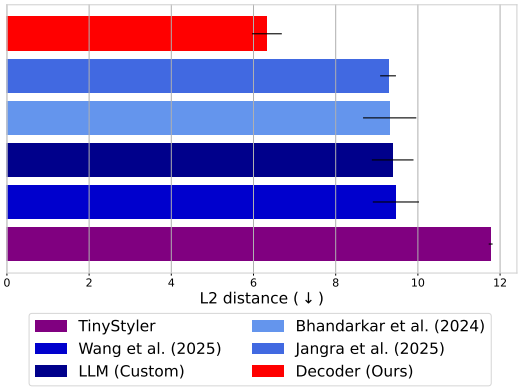
(b) StyleDistance

Figure A2: Our approach yields the smallest L2 distance, measured by other style representations too.

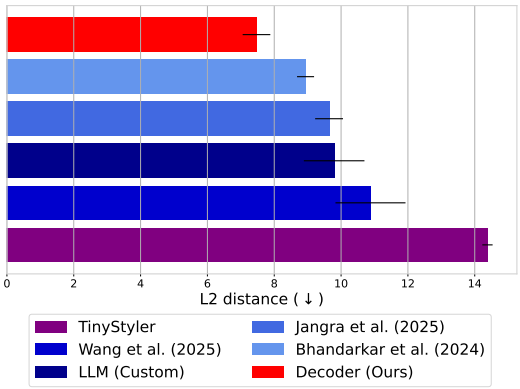
G Supplementary Material for Section 8

Figure A3 shows the evaluation in Figure 5 repeated using two alternative style representation models: LUAR (Rivera-Soto et al., 2021) and StyleDistance (Patel et al., 2025). Table A2 presents all numeric values for Figures 5 and A3.

Again, the trends across different representation models align, demonstrating that our decoder is not limited to the style representation used for training.



(a) LUAR



(b) StyleDistance

Figure A3: Our decoder outperforms baselines at generating in human-like styles, measured by other style representations too.

Method	Our Embedding	LUAR	StyleDistance
Decoder (Ours)	27.73	6.33	7.47
LLM (Custom)	37.54	9.39	9.79
Wang et al. (2025)	39.28	9.46	10.88
Jangra et al. (2025)	42.01	9.27	9.64
Bhandarkar et al. (2024)	35.53	9.31	8.94
TinyStyler	54.69	11.77	14.38

Table A2: The numeric values for evaluation in Section 8.

Model	Num. Params.	Huggingface Link
Phi-4	14.7B	microsoft/phi-4
Qwen2.5-14B	14.8B	Qwen/Qwen2.5-14B
OLMo-2-13B	13.7B	allenai/OLMo-2-1124-13B
Mistral-Nemo-Instruct-2407	12.2B	mistralai/Mistral-Nemo-Instruct-2407
Ministral-8B-Instruct-2410	8.0B	mistralai/Ministral-8B-Instruct-2410
Llama-3.1-8B-Instruct	8.0B	meta-llama/Llama-3.1-8B-Instruct

Table A3: The LLMs used in our study.

Category	# features
Writing Goal and Intent	40
Tone and Mood	40
Readability Level	40
Formality	40
Narrative Perspective and Voice	40
Social and Interpersonal	40
Audience Engagement and Interaction	40
Emotional Intensity	38
Descriptive Density	40
Information Density	40
Logical Structure and Flow	39
Creativity and Typicality	39
Abstraction Level	39
Temporal Focus	39
Technical and Domain Specificity	40
Cultural and Regional Influences	39
Figurative Language Usage	38
Rhetorical Device Usage	38
Intertextuality and Allusion	38
Sentence Structure and Syntax	38
Visual Formatting and Layout	39
Function Word Usage	38
Word and Expression Choice	39
Punctuation Usage	39
Special Character and Capitalization Usage	31
Acronym and Abbreviation Usage	39
Total	1,010

Table A4: The full list of 26 stylistic categories on which we curate style features.

Topic	Platform
Advice	Reddit
AmItheAsshole	Reddit
AskMen	Reddit
AskReddit	Reddit
askscience	Reddit
AskWomen	Reddit
explainlikeimfive	Reddit
NoStupidQuestions	Reddit
relationship_advice	Reddit
relationships	Reddit
academia	StackExchange
cooking	StackExchange
diy	StackExchange
history	StackExchange
law	StackExchange
money	StackExchange
philosophy	StackExchange
politics	StackExchange
scifi	StackExchange
workplace	StackExchange
business and finance	Yahoo Answers
computers and internet	Yahoo Answers
education and reference	Yahoo Answers
entertainment and music	Yahoo Answers
family and relationships	Yahoo Answers
health	Yahoo Answers
politics and government	Yahoo Answers
science and mathematics	Yahoo Answers
society and culture	Yahoo Answers
sports	Yahoo Answers

Table A5: The full list of 30 topics from which we source our questions.