What is it like to write like Haiku?

Jinran Jin, Ryan Teehan, Jack Lu, and Mengye Ren
Agentic Learning AI Lab, New York University
{jj3007, rst306, yl11330, mengye}@nyu.edu
https://agenticlearning.ai/prompto
November 12, 2026

Abstract

Language models have distinct and recognizable writing styles, yet these writing styles are rarely articulated precisely. In this paper, we study whether a model's style can be described in words for another model to adapt and follow. To measure the fitness of a description, we ask independent evaluators to distinguish the pair of responses. Rather than coming up with an *ad hoc* prompt, our proposed framework, PrompTo, creates a search loop to optimize for the style prompt that confuses the evaluator. A reflector then takes in the reward and language feedback and generates a prompt candidate for the next iteration. The optimized prompts recover 74.8% of the source–target gap on average, against at most 15.2% for existing baselines. Moreover, we found style prompts optimized with one source model readily transfer to other source models.

1 Introduction

In August 2025, GPT-5 abruptly replaced all prior models in ChatGPT. Within days, users started campaigning for the return of GPT-4o (Lai, 2026), citing GPT-4o's "warm" tone as the quality they missed most. While GPT-4o may in fact write more warmly than GPT-5, such a description is not enough to fully articulate its style, and simply telling GPT-5 to write more warmly would not make the experience of interacting with it comparable. What we lack is a description of GPT-4o's style that is comprehensive enough that another model could follow it and sound like GPT-4o. Applied to models more generally, we say that a description of a model's style is complete if a different model that has not seen the original model's writing can reproduce its style from the description alone. We ask whether such a description of models' writing styles can be found, so that another model can follow it to emulate its writing.

Prior work on LLM style has focused on distinguishing between the styles of different LLMs and extracting specific natural language characteristics that describe their pairwise differences, but such descriptions have not been tested with reproduction. Sun et al. (2025) showed that classifier trained on model responses can identify which model authored a response with 97.1% accuracy. VibeCheck (Dunlap et al., 2025) uses pairwise comparisons to extract explicit and human-readable axes, in natural language that describe the differences between model outputs. Though these characteristics are interpretable, and can be used to classify models, they do not allow for reproducing a particular model's style.

Agentic Learning AI Lab Technical Report AGL-TR-2026-03. doi:10.5281/zenodo.24160383

By contrast, work in text style transfer allows for style reproduction but does so without interpretable natural language descriptions. Conditioning a model on an author's writing samples or fine-tuning on the author's text, for example, can produce output in that author's style (Wang et al., 2025; Mysore et al., 2024; Gudibande et al., 2023), but the style is never described explicitly.

No prior work both describes a model's style and reproduces it. We close this gap with **PromptTo**, a framework that searches for a complete description of a target model's style and tests it by reproduction. We capture this description in a system prompt, which we call a style prompt. To test the style prompt, we prompt a source model to generate restyled text, and use a discriminator to assess the likeness between the text generated by target model and the restyled source model output. We report how far the style prompt moves the discriminator toward chance as Style Gap Recovered (SGR). An SGR of 0% means the discriminator separates restyled responses from target responses just like it separates unprompted source responses from target responses. An SGR of 100% means the discriminator can no longer separate the restyled responses from target responses. To search for a complete style prompt, we pair the discriminator with a reflection model, which we call a reflector. The discriminator's reasoning is passed to the reflector, which revises the style prompt, as shown in Figure 1. To assess the style prompts, we consider three different kinds of evaluators: an LLM-as-judge discriminator, a pairwise authorship-verification classifier, and a style encoder trained using a contrastive loss. The search process requires no access to either model's weights, so proprietary models can serve as any roles.

We apply our framework to three target models—Gemma, Qwen, and Claude Haiku—and find that each has a style that is distinct, describable in natural language, and stable across responses. From the style prompts, we find actionable mechanisms behind the styles. For example, Gemma is an enthusiastic tutor because it uses warm openings and asks follow-up questions, Qwen is a neutral assistant because it goes directly to the question and does nothing else, and Haiku is a knowledgeable friend because it gets straight to the answer with warm voice and no AI-like phrases. We also find that style prompts optimized for a specific source and target model can be used to transfer the style of other source models to the same target model with most of the SGR retained. This indicates that style prompts mainly describe traits of the target's style that should be emulated rather describing traits of the source model's style that should be suppressed. Finally, we find that the style prompts that PromptTo generates have a similar structure and each target model is distinguished by a different part of it. We call each combination of source model, target model, and dataset a direction. Across 13 directions on 2 datasets, the optimized style prompts recover 74.8% of the source-target gap on the discriminators, against at most 15.2% for any baseline.

2 PromptTo for Stylistic Transfer

PromptTo searches for a style description in the form of a system prompt called style prompt π that makes a source model M_s write like a target model M_t (Figure 1). In each iteration, we generate restyled responses under π, evaluate π with an LLM discriminator, and propose an updated π with reflector using the discriminator's reasoning and failure cases.

2.1 Discriminator

An off-the-shelf LLM plays the role of discriminator D, which evaluates the style prompt through in-context learning. For each question, D receives the restyled response Y_s' from M_s under π, the target response Y_t from M_t answering the same question, and n anchor responses Y_a from M_t. D selects the response closer in style to the anchors. Y_a is the answer to different questions from Y_t, since otherwise D could select Y_t by comparing content instead of styles. The same target model is very likely to answer

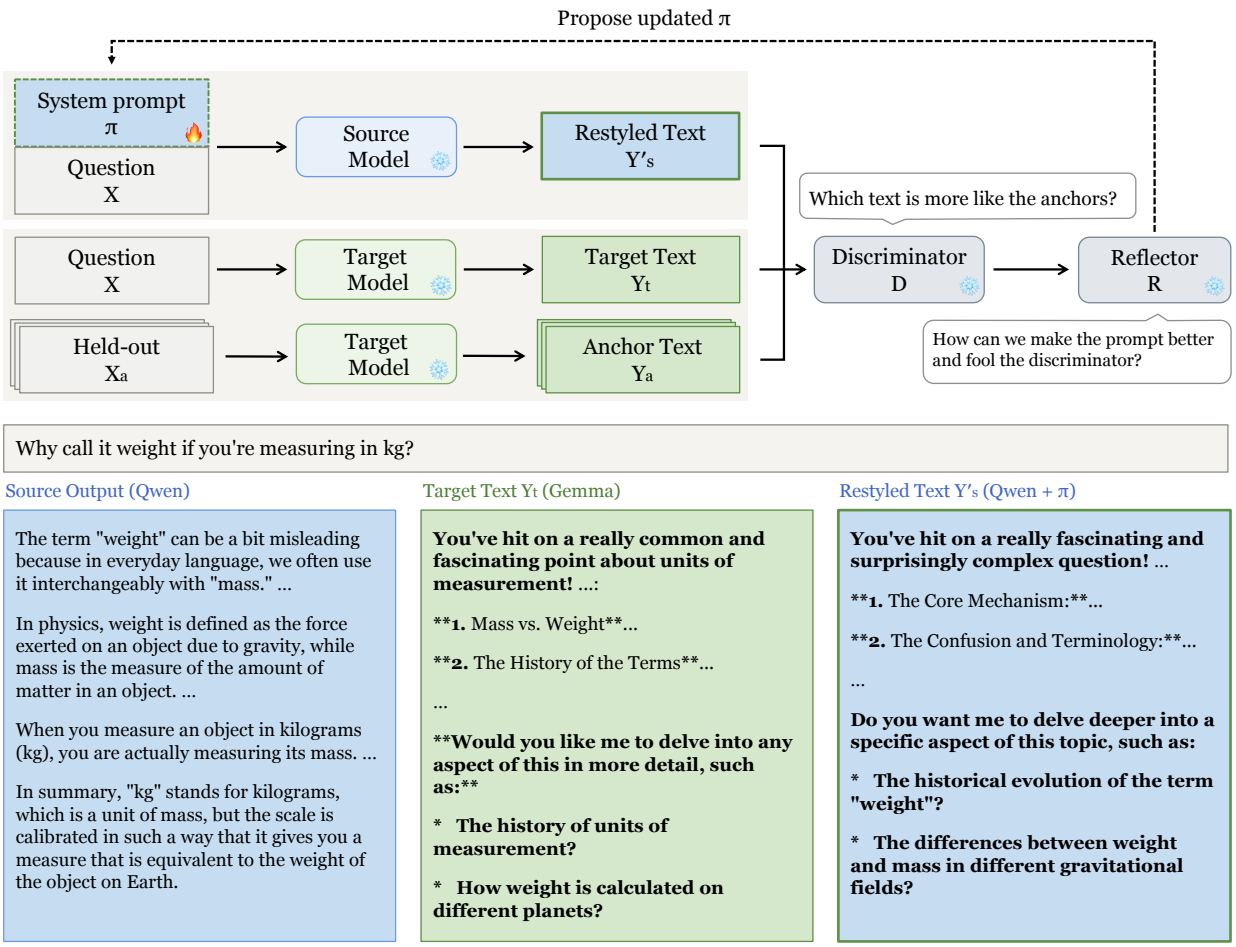


Figure 1: ***The PrompTo framework for stylistic transfer.*** *Top: the source model answers X under a trainable style prompt π to generate the restyled text Y_s'. The target model answers X and held-out X_a to generate Y_t and Y_a. Discriminator D picks which of Y_s', Y_t is closer in style to Y_a. The reflector R proposes updated π based on D's reasoning. Bottom: Qwen2.5-7B → Gemma3-4B on ELI5. Bold text shows the style of the target model.*

the question with similar content. To control the positional bias of the LLM discriminator, we randomize the order of Y_t and Y_s' in each trial. We measure how well π reproduces the target's style by its fool rate, defined as the probability that D predicts that Y_s' is more similar to Y_a than Y_t is. A fool rate of 50% means the restyled responses are almost identical in terms of style to the target responses.

2.2 Prompt Search

The search for a style prompt starts from a generic seed prompt with no stylistic content ("*You are a helpful assistant that can answer questions and help with tasks.*"), which tends to have a fool rate near zero. Any prompt optimizer can be used in this search process, but we use GEPA (Agrawal et al., 2026) in our experiments because it takes in natural language feedback. This feedback makes the optimization signal rich. GEPA is a reflective prompt evolution algorithm which maintains a pool of candidate prompts and iteratively mutates them using a reflection model. At each step of the search process, PrompTo generates

a batch of restyled responses Y_s' with the source model M_s under the current style prompt π, evaluates π by asking an LLM discriminator to predict which of the responses Y_s' and Y_t were generated with M_t, and uses an LLM reflector R to propose a new π after seeing the response pairs that D correctly judged along with D's reasoning. Only the style prompt changes, with M_s, M_t, D, and R stays frozen. The objective is to find a style prompt that maximize the fool rate which can be written as:

$$\pi^* = \arg\max_{\pi} \text{FoolRate}(\pi)$$

2.3 Evaluation

We evaluate our style prompt π^* using both the LLM discriminator used during training as well as four held-out evaluators, including two LLM discriminators which not used for prompt optimization and two non-LLM evaluators.

Discriminators. We report the fool rates of the Qwen3.8-27B discriminator, which is used during GEPA optimization, and two discriminators that never appear during the optimization (Qwen3.6-35B-A3B and gpt-oss-120b). Each discriminator is evaluated with the same protocol described in Section 2.1. We report the averaged fool rate for the three discriminators.

Authorship verification. We train a pairwise authorship-verification (AV) classifier following Wang et al. (2025) and using a ModernBERT-base encoder (Warner et al., 2024). The encoder receives two responses and must predict whether they were generated by the same model. Training pairs are built from raw responses without style prompts of all five models, and balanced between same-model and different-model pairs. At test time we present the model with pairs of restyled and target response generated for the same question and report the fraction of pairs that are classified as being generated by the same model, which we call the AV ratio.

Style embeddings. We use STAR (Huertas-Tato et al., 2023), a contrastively trained style encoder, to compute how similar our restyled responses are to the anchor responses. For each test question, we embed the n anchor responses Y_a from M_t and average their embeddings to produce a target centroid. We then compute the cosine similarity of the restyled response to the target centroid.

Style gap recovered. The 13 transfer directions shown in Table 1 vary in difficulty, because some model pairs are already similar in writing styles without any style prompt, so raw fool rates are not directly comparable across them. Therefore, we report style gap recovered (SGR), which measures how far restyling moves the fool rate from its floor to its ceiling:

$$\text{SGR} = \min\left\{1, \frac{F(\text{restyled}) - F(\text{source})}{0.5 - F(\text{source})}\right\},$$

where $F(\cdot)$ is the fool rate of an LLM discriminator. We compute SGR separately for each discriminator and direction, then average them. $F(\text{source})$ is measured on unprompted source response, and $F(\text{restyled})$ is measured on source response under π^*. The ceiling is fixed at 0.5, which represents discriminator guesses at chance. SGR = 0% means the restyled responses are same as raw source responses. SGR = 100% means the discriminator can no longer tell the difference between the restyled and target. Because for some direction, LLM discriminator tends to give fool-rate being higher than 0.5, we cap SGR to 1.

3 Experiments

3.1 Experimental Setup

We evaluate whether the style prompts that PrompTo produces are complete enough to make a source model's responses indistinguishable from those of a target model and compare our approach to simpler methods of generating descriptions for style transfer. Our experiments span 13 direction-dataset configurations built from 5 models and 2 question datasets, as shown in Table 1.

Datasets. We evaluate our method by generating responses to free-form questions, because the responses leave room for stylistic variation to emerge. Specifically, we use questions from the ELI5 (Fan et al., 2019) dataset, which asks for explanations of everyday questions from the r/explainlikeimfive subreddit, and Chatbot Arena Conversations (Zheng et al., 2023) dataset, which contains user prompts from pairwise model battles. For Chatbot Arena Conversations, we filter the dataset to keep only the first turn of the conversations. We sample 8725 questions for training and 2000 questions for testing from each dataset.

Models. For source and target model, we use Qwen2.5-3B, Qwen2.5-7B (Qwen et al., 2025), Gemma3-4B (Team et al., 2025), Llama3.2-3B (Grattafiori et al., 2024), and Claude Haiku 4.5 (Anthropic, 2025) in our experiments. The four open-weight models serve as both source and target models, while Claude Haiku 4.5 serves only as a target model for budget reasons. We use Qwen3.8-27B as both the reflector and discriminator model. We only retain direction-dataset configurations whose baseline discriminator accuracy on original responses is above 75%. Below that, the evaluation would be unreliable because the model responses are already hard to distinguish, and the training signal would be poor. With this filtering, we have 13 configurations. Most excluded configurations are models with similar styles (e.g. Qwen and Llama) or datasets with questions that are so diverse that they make it hard for discriminators to judge.

Dataset	Source	Target(s)
ELI5	Gemma3-4B	Claude Haiku 4.5, Llama3.2-3B, Qwen2.5-3B, Qwen2.5-7B
	Llama3.2-3B	Claude Haiku 4.5, Gemma3-4B
	Qwen2.5-3B	Claude Haiku 4.5, Gemma3-4B
	Qwen2.5-7B	Claude Haiku 4.5, Gemma3-4B
Chatbot Arena	Llama3.2-3B	Gemma3-4B
	Qwen2.5-3B	Gemma3-4B
	Qwen2.5-7B	Gemma3-4B

Table 1: *Source-target directions covered in our evaluation.*

Baselines. We have four baselines for comparison, each of which is a different kind of description of the target's style. We do not compare with VibeCheck (Dunlap et al., 2025), since it does not give an actionable description for transferring the style of one model to another.

- **Vanilla:** We give the source model no description to follow. This baseline measures how distinguishable the two models are out of the box, and its score is the floor of the SGR.
- **Generic:** We only give the source the target model's name (e.g. "Write in the style of Qwen"). This tests whether the source model can emulate the target model using information already learned during pretraining.
- **Few-shot:** (Wang et al., 2025) In lieu of a description, we give the source model three target responses answering other questions. This provides an implicit description of the style captured through in-context examples, and tests whether the source model can learn from examples to follow the style consistently.
- **Single-pass:** We give the reflector the same 10 target responses that PrompTo starts from, and ask it to generate a style description. The reflector is not allowed to revise this description. This tests the contribution of discriminator-driven prompt optimization.

Table 2: ***Main results of PromptTo****, averaged over the 13 directions across three kinds of evaluators. Both the Fool rate and its Style Gap Recovered (SGR) are averaged across three LLM discriminators. The AV ratio is the AV classifier's same-model rate. For "Cos. sim.', we compute the STAR embedding cosine similarity between the text and centroids. Higher is better throughout.*

Method	Fool rate (%)	Fool rate SGR (%)	AV Ratio (%)	Cos. sim.
Vanilla	9.2	–	0.9	0.767
Generic	3.5	−21.3	6.1	0.692
Few-shot	14.5	15.2	10.3	0.772
Single-pass	16.6	13.8	19.5	0.769
PrompTo (ours)	**43.8**	**74.8**	**51.0**	**0.801**

3.2 Main results

Table 2 shows the averaged results over the 13 directions listed in Table 1. PrompTo's optimized style prompts close the source-target gap more than any other kind of description. The PrompTo style prompt recovers 74.8% of the source–target gap on average when measured by LLM discriminators, compared with only 15.2% for Few-shot, the strongest baseline. The results from non-LLM evaluators, AV Ratio and STAR embedding cosine similarity, similarly rank PrompTo's style prompt as the best, indicating that the optimized style prompts consistently capture properties of target model style visible to different kinds of evaluators. Among the baselines, Generic is the worst with an average fool rate of 3.5%. Telling a source model to "write like Qwen" makes the output less Qwen-like, indicating the source model does not have enough knowledge of the style of other models to reliably reproduce their writing styles. Few-shot and Single-pass both have modest, and similar, Fool Rate and SGR. We find that Few-shot produces outputs with better style embedding cosine similarity, while Single-pass produces outputs that fool the AV classifier more often. In summary, we find that a model name alone does not carry much style information while in-context examples and un-revised style descriptions carry some style information, but not enough for reproduction. Only PrompTo's style prompt, which is optimized with a discriminator, carries enough information to reproduce the target model's style consistently.

3.3 Variation across configurations.

The average in Table 2 hides a wide spread for each model and dataset configuration. PrompTo's fool rate under the Qwen3.8-27B discriminator ranges from 9.3% to 81.0%. Figure 2 (a) shows how source model or target model affect the fool rate. Evaluation results for all directions separately are shown in SectionA.3

Source model. We first study the effect of the source model on style emulation performance. With the target fixed as Claude Haiku 4.5 or Gemma 4B on ELI5 (Figure 2a), the initial fool rates are nearly identical across the different source models. Once each π^* is applied, however, we see a divergence between their rates. Using Qwen 7B as a source model gives the highest fool rate for both target models, compared to the fool rates from the two 3B source models. This ordering shows that the capacity of the source model may be important in style emulation. We find that the gap between fool rates is mainly due to how well each source model follows the style prompts. For example, the style prompts targeting Haiku specify the same features regardless of the source model, but Llama3.2-3B and Qwen2.5-3B follows the directions of the style prompt less reliably. Weaker models also tend to misread instructions. Qwen2.5-3B's style prompt shows "Main reason" and "Practical takeaway" as examples of possible section titles, but the model treats

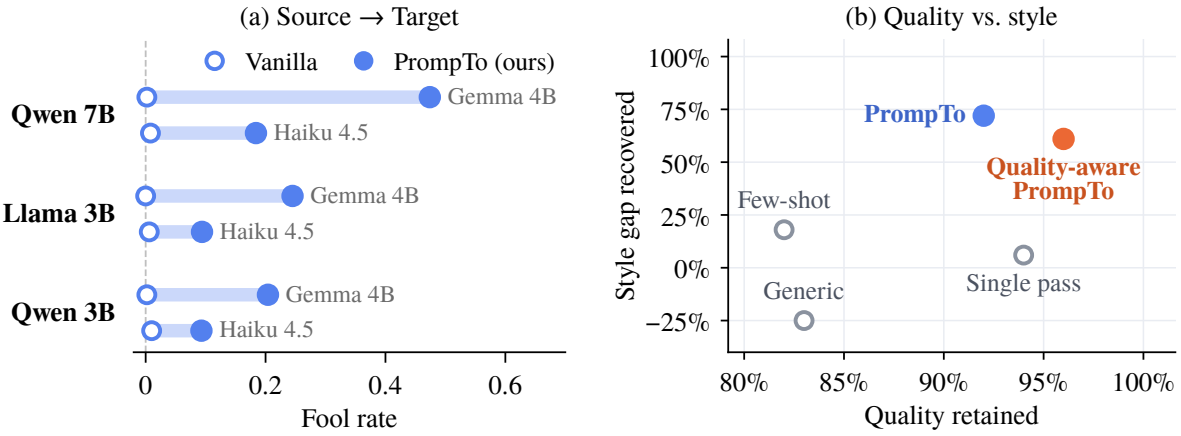


Figure 2: ***(a) Effect of source and target model.*** *For each source, we vary the target between Gemma3-4B and Claude Haiku 4.5. The final fool rate depends on both the source model's capacity and target model's difficulty.* ***(b) Quality vs. style.*** *We compare each method in terms of both their SGR as well as how much quality they retain after applying π^*. We see Quality-aware PromptTo recovers the quality drop caused by PromptTo, but recovers less style gap. Quality retained is the quality of the restyled response as a fraction of the raw response's quality.*

them as a fixed template and reuse them in 99% of its answers, giving every response the same structure where Haiku's headings change with the question and content.

Target model. Next, we investigate how the target model impacts style emulation. Using the same set of source and target models described above, we find that every source model finds Gemma's style easier to describe and execute than Haiku's. The gap is largest for Qwen2.5-7B, whose fool rate reaches near chance when targeting Gemma but only 18% when targeting Haiku. Llama3.2-3B and Qwen2.5-3B show the similar trend, with fool rates when targeting Gemma that are nearly double the fool rates when targeting Haiku. The opening of the target model's response plays an important role here. Gemma tends to open with a small set of fixed phrases that a style prompt can quote directly. On the other hand, Haiku's openings are more diverse and content-dependent, which are harder for a smaller model to learn. This leads to a lower fool rate.

3.4 Style vs. Quality

Style transfer can come at a cost to response quality. To quantify this cost, we compare the quality of raw source responses against responses generated under π^* on the 10 ELI5 directions. Because the questions are open-ended, we use an LLM-as-judge. Qwen3.6-35B-A3B scores each response independently on a 1–10 scale, and we report the quality drop, defined as the mean score of raw source responses minus that of restyled responses. On average, PromptTo drops quality by 0.45 points, roughly half the drop of the Generic and Few-shot baselines.

To alleviate the quality drop, we design a quality-aware variant of PromptTo. In the original PromptTo, the reflector receives the questions and responses that fail to fool the discriminator. The quality-aware variant additionally receives the restyled responses with the largest quality drop relative to the raw source responses, and is explicitly prompted to prevent such quality drop while restyling. For simplicity, these examples are added to the reflector's prompt and we keep only fool rate as the GEPA objective.

Table 3: *Stylistic features of target models.* *Frequencies counted with regex.*

	Frequency in model output		
Stylistic feature	Gemma-3-4B	Qwen-2.5-7B	Haiku 4.5
Warm opener ("You've hit on…", "That's a fantastic…")	**80%**	<1%	<1%
Bold numbered section headers `**1. Title:**`	**87%**	0%	1%
"Resources for further…" section with links	**73%**	<1%	<1%
Closing menu ("Do you want me to delve deeper…?")	**87%**	<1%	<1%
"Here are some / a few…" lead-in	2%	**30%**	2%
Flat numbered list, colon outside bold `1. **Label**:`	<1%	**74%**	2%
"In summary" closer	3%	**30%**	0%
`#` title on the first line	0%	0%	**73%**
`##` section headers	0%	0%	**60%**
Hyphen bullets `-`	0%	25%	**89%**
Em dash (—; per 1k words)	0.1	0.1	**8.6**

As shown in Figure 2 (b), the quality-aware variant roughly halves the quality drop, but with the trade-off of a lower fool rate. The quality-aware style prompts contain instructions that never appear in the original ones, such as "Your first job is to answer the user's actual question accurately and completely." These instructions make the source model prioritize content over style. The quality-aware prompts also drop some strict rules, such as forcing a resources section when targeting Gemma. The resulting style prompts are more generic and content-first, which increases quality but decreases the fool rate.

4 Style Prompt Study

4.1 Model Signature

We the optimized style prompt for three different target models, Qwen2.5-7B, Gemma3-4B, and Claude Haiku 4.5 and find that they each have a namable persona. PromptTo's style prompts describe Gemma as an enthusiastic tutor because it uses warm openings and asks follow-up questions, they describe Qwen as a neutral assistant because it goes directly to the question and does nothing else, and they describe Haiku as a knowledgeable friend because it gets straight to the answer with warm voice but no AI-like phrases. For each model, we measure the frequency of their distinctive features with regex. All of the features discussed below appear in their style prompts, and are shown in Table 3 with associated frequencies. We do not include Llama3.2-3B in our analysis because we only have one direction where it is the target model, and its style is similar to Qwen's.

Gemma Every π^* that targets Gemma opens by describing it as "enthusiastic," "warm," and "conversational" regardless of the source model. Gemma's responses almost always begin with a warm opening, and end with follow-up questions. These characteristics give the impression of a model that enjoys the conversation and wants to keep it going even after the question is answered. The most distinctive feature the π^*s recover is the closing phrases. "Delve deeper" appears in 87% of Gemma's ELI5 responses and

Table 4: *Cosine distance between π^*.* *We compare three groups of π^* with Qwen3-Embedding-8B. Distance shows sharing a target pulls two π^* together.*

Group	Mean distance
Same target, different source	**0.212**
Same source, different target	**0.343**
Different source, different target ("unrelated")	**0.348**

almost never in those of other models. Warm openers together appear in 80% of Gemma's response across both datasets and also almost never in those of other models. According to the discriminator's reasoning, these two sets of phrases are what distinguishes Gemma from the other models.

What the style prompts capture but the source models fail to execute is the distribution over these phrases. Gemma's style is a distribution over a small set of fixed phrases, and the style prompt captures this uncertainty, but the restyled responses collapse the distribution to only one or two phrases. However, the mismatch is difficult for the discriminator to detect, because we only use it one at a time with no memory, which is why the fool rate in this direction remains high.

Qwen Qwen's style prompts do not contain personality terms like the other two models. "Neutral" and "plain" are the two most appeared terms across the style prompts that target Qwen. The style prompts also spend more effort on what to avoid comparing to the style prompts of other models.

Qwen's signature is largely absent. It never asks a question back, provides links, or writes disclaimers. It does not have a distinctive opening and its most common opening phrase only covers 0.2% of its responses while Gemma's covers 27.5%. The unique feature of Qwen's style is the format of the body of its responses. Most of Qwen2.5-7B's responses are a single flat numbered list. The fact that Qwen's responses are plain makes them easier to emulate, and among the three target models Qwen yields the highest SGR. When Gemma is used as source model targeting Qwen, it only needs to drop the opening and closing phrases and write the body as a numbered list to emulate Qwen's style. In the directions we study, removing features is easier for a small model than adding them.

Haiku Most style prompts that target Haiku ask the source model to be slightly conversational, to sound like a knowledgeable person or friend, and to not be as formal as an encyclopedia. This places Haiku's style between Gemma's and Qwen's. Haiku's most distinctive feature is the title, as most of its responses open with a Markdown heading, which no other models do. Each style prompt targeting Haiku specifies how to write such a heading, and they even cover how specific the title should be, what to avoid, and often give a few examples. The π^*s describe the structure of the response as a short document.

Haiku's responses are hard to emulate because it is much more capable than the source models we consider. Once the style matches, the remaining differences are the depth and correctness of the answer. No 3–7B model can close this gap with a single system prompt. These differences are also effect the discriminators. gpt-oss-120b's judgements, for example, focus on surface level styles, therefore it is often fooled when targeting Haiku. However, Qwen3.8-27B tends to be a more capable LLM style discriminator and is fooled less often.

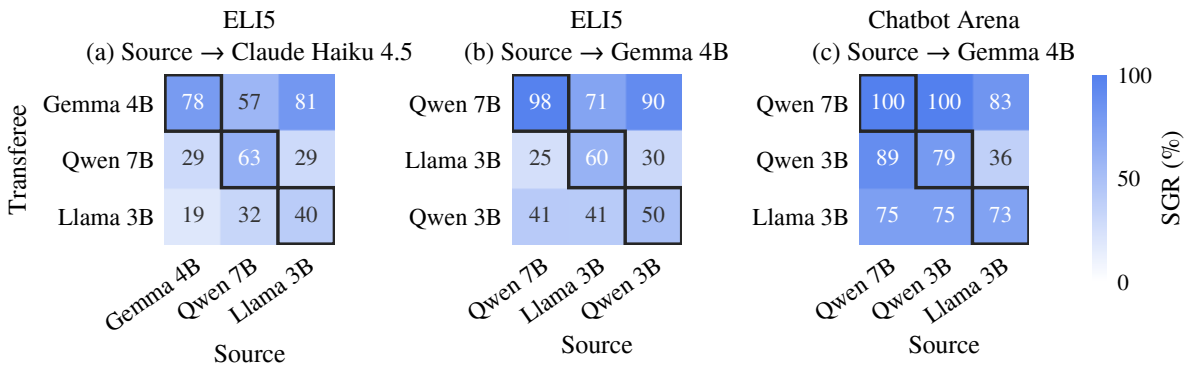


Figure 3: ***Style prompts transfer across source models.*** *Each panel is one target on one dataset. Each row is a source model executing a style prompt and each column is the source model the prompt was optimized for. Boxed diagonal cells are each model running its own prompt and off-diagonal cells are transferred prompts.*

4.2 Style Prompt Generalization

To determine whether π^* describes the target's style or the difference in style between a particular source and the target, we swap the style prompts between different source models for a fixed target. For a style prompt that transfers the style of a source model to the style of a target model, we call the source model the owner of the style prompt because it was optimized for that source model specifically. Any other source models running the style prompt are called the transferee. If π^* mainly describes the target's style, we should see that the transferee can still achieve target-like style despite using a prompt optimized for a different model. We find that transferees retain 78% of the SGR on average compared to the owner (Figure 3). We additionally find that swapping style prompt does not hurt as much as swapping to a weaker source model, for example Gemma3-4B running with transferred prompt still beats Llama3.2-3B running on its own prompt (Figure 3 (a)).

Text embeddings provide further evidence that each π^* mainly describes the target model's style. We use Qwen3-Embedding-8B to embed each π^* and compare cosine distances between three groups of pairs: different source models with same target model, the same source model with different target models, and different source models with different target models. Results in Table 4 shows that style prompts that share a target have higher cosine similarity than those that share a source model, and the latter are nearly as dissimilar as style prompts that share neither a source nor a target model. This further supports the claim that π^*s describe the target model's style rather than the difference between the source model and target model's styles.

4.3 Ablation

To see which parts of the π^* actually carry the target style, we ablate components of π^* for the Gemma3-4B and Qwen2.5-7B pair in both directions. We pick this pair because both directions reach high SGR, which leaves room to show the effects of the ablations. We manually partition each π^* into the six categories shown in Table 5. Both style prompts share the same structure, but they each spend more effort on different things. The Gemma-target style prompt spends more on voice and closing, whereas the Qwen-target style prompt spends more on body structure. For the lines that describe two categories at the same time, we ablate it for both categories to ensure no leakage. The way we categorize is shown in Table 7.

Table 5 shows that feature importance is target-specific. For Gemma3-4B, removing the opening, closing

Table 5: *SGR of different ablations across categories in π^*.* *We do ablations in two directions, using Qwen and Gemma models be the source or target model. The category that contributes to most SGR drop is different for the two models. Gemma suffers more from ablating voice, opening, and closing, while Qwen suffers more from ablating body structure.*

		Category removed					
Direction	Full	Voice	Opening	Structure	Content depth	Closing	Formatting
→ Gemma3-4B	94.8	34.2	15.2	47.8	41.4	20.8	98.8
→ Qwen2.5-7B	100.0	100.0	69.8	44.7	100.0	100.0	100.0

and voice hurts SGR most, while removing content depth or structure costs less SGR. For Qwen2.5-7B, body structure becomes the most important category, and dropping voice or closing barely makes a difference. In both directions, the categories that cost the most correspond to the most distinctive features identified by frequency in Section 4.1. In addition, not everything in the style prompt contributes much to SGR. Formatting is described in both style prompts, but removing it barely moves SGR. All of these rankings are stable across the three LLM discriminators with $\rho \geq 0.94$ within each direction.

4.4 Feature Frequency Mismatch

Fool rate being above 50% reveals potential feature frequency mismatch. Intuitively, a fool rate above 50% means the LLM discriminator predicts the restyled response as being more similar to the anchors comparing to the actual target response. This is caused by frequency mismatch shown in Table 6. The model's behavior can be seen as a distribution of phrases, but the restyled text collapse it into just one behavior. Most of the options are present in π^*, meaning the cause of the collapse is mainly due to the source model unable to pick choices stochastically. The style prompt provides a list of openers, but the source model only pick one of them, and it picks it almost every time. Because of the way GEPA optimize the style prompt, it does not penalize the style prompt for not making distribution diverse. For discriminator model, it only sees one anchor, target, restyled triplet at a time, therefore it cannot see the frequency mismatch either. This leads to distribution collapse. The collapse then caused the discriminator to find the restyled response being more consistent to the anchor comparing to the original target text.

5 Related Work

5.1 Style Transfer

Text style transfer work falls into two regimes. In the first, the target style is a predefined attribute with a name. Prior work in this regime transfers a single such attribute, e.g., formality (Rao and Tetreault, 2018), politeness (Madaan et al., 2020), or sentiment (Li et al., 2018). With the advancement of LLMs, such attribute-level restyling can be performed by prompting alone (Reif et al., 2022; Suzgun et al., 2022; Pan et al., 2024). However, mimicking a specific author requires more than named attributes, because individual style is more nuanced than predefined labels can capture (Kang and Hovy, 2021). Work in the second regime therefore learns such implicit styles directly from an author's text. Few-shot approaches condition on a handful of the author's writing samples (Wang et al., 2025; Bhandarkar et al., 2024). Retrieval-based systems select stylistically relevant examples per query (Mysore et al., 2024; Li et al., 2023b,a). Fine-tuning

Table 6: ***Feature frequency for target model vs. source model with π^*.*** *We see original target model tends to write in a more diverse way, but source model with π^* failed to replicate that diversity. Because the LLM discriminator is only shown with one test at a time, it cannot spot the distribution mismatch.*

Style choice	Option	Frequency in model output	
		Gemma-3-4B	Qwen-2.5-7B restyle
Gemma opener	"You've hit on…"	38%	99%
	"Okay, …"	34%	0%
	direct answer	19%	1%
	"That's a fantastic…"	8%	<1%
		Qwen-2.5-3B	Gemma-3-4B restyle
Qwen-3B list layout	numbered list with bold labels	69%	100%
	prose only	18%	0%
	#/## sections	9%	0%
		Qwen-2.5-3B	Gemma-3-4B restyle
Qwen-3B closer	plain statement	59%	0%
	"In summary / In conclusion…"	31%	96%
	caveat ("important to note…")	9%	3%

approaches adapt model weights on the author's corpus (Zhang et al., 2018).

5.2 Authorship Analysis

Authorship analysis asks who wrote a text. The problem in authorship analysis can be formulated as attribution, which selects an author from a closed candidate set, or verification, which decides whether two documents share an author (Stamatatos, 2009; Koppel et al., 2009). Classical methods represent a document by hand-crafted stylometric features (Burrows, 2002; Stamatatos, 2009). Neural methods replace hand-crafted features with learned representations. Contrastive authorship encoders (Rivera-Soto et al., 2021; Huertas-Tato et al., 2023). Supervised pairwise classifiers built on pretrained encoders instead and learn the decision directly (Boenninghoff et al., 2019).

5.3 Prompt Optimization

Methods for automatically improving prompts differ in what they require of the model. Continuous prompt tuning learns a soft prefix by gradient descent through the model's weights (Li and Liang, 2021; Lester et al., 2021). Another body of work treats the model as a black box and searches over natural-language instructions, using an LLM to propose, critique, and recombine candidates (Zhou et al., 2023; Yang et al., 2024; Pryzant et al., 2023; Guo et al., 2025; Khattab et al., 2023). GEPA is a recent work, which uses reflective natural-language feedback and Pareto-aware selection and reports outperforming GRPO with far fewer rollouts. We build on GEPA because it requires no access to model weights. And most importantly, the result is an optimized text, which makes it possible to learn model's implicit styles in an explicit system prompt.

6 Conclusion

Learning the style of language models is important, because it may explain why users prefer one model over another beyond capability. Learning it in an interpretable way allows people to know more about a model that otherwise seems like a black box. In this work, we find a model's implicit style can be written down and optimized in the form of a system prompt, which can be used by other models to reproduce the style. To this end, we designed evaluations and metrics and show that PrompTo is much more effective in style transfer than the standard practices of using exemplar prompting or single-pass orchestration. Currently, all of our source models in our experiments are small and open-weight, but the technique can scale easily. In future work, we hope to expand the study to more models, and collect human preference data on LLM writing styles to build models with styles users prefer.

Acknowledgement

This research was supported by Visko Platform, Pionex, a Google TPU Award, a Google ML and Systems Junior Faculty Award, and the Institute of Information & Communications Technology Planning & Evaluation (IITP) under grants RS-2024-00469482 and RS-2024-00509279, funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research. The compute is supported by the NYU High Performance Computing resources, services, and staff expertise.

References

Agrawal, L. A., Tan, S., Soylu, D., Ziems, N., Khare, R., Opsahl-Ong, K., Singhvi, A., Shandilya, H., Ryan, M. J., Jiang, M., Potts, C., Sen, K., Dimakis, A. G., Stoica, I., Klein, D., Zaharia, M., and Khattab, O. (2026). Gepa: Reflective prompt evolution can outperform reinforcement learning.

Anthropic (2025). System card: Claude Haiku 4.5.

Bhandarkar, A., Wilson, R., Swarup, A., and Woodard, D. (2024). Emulating author style: A feasibility study of prompt-enabled text stylization with off-the-shelf LLMs. In Deshpande, A., Hwang, E., Murahari, V., Park, J. S., Yang, D., Sabharwal, A., Narasimhan, K., and Kalyan, A., editors, *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 76–82, St. Julians, Malta. Association for Computational Linguistics.

Boenninghoff, B., Hessler, S., Kolossa, D., and Nickel, R. M. (2019). Explainable authorship verification in social media via attention-based similarity learning.

Burrows, J. F. (2002). 'delta': a measure of stylistic difference and a guide to likely authorship. *Lit. Linguistic Comput.*, 17:267–287.

Dunlap, L., Mandal, K., Darrell, T., Steinhardt, J., and Gonzalez, J. E. (2025). Vibecheck: Discover and quantify qualitative differences in large language models.

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M. (2019). Eli5: Long form question answering.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A.,

Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal,

M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. (2024). The llama 3 herd of models.

Gudibande, A., Wallace, E., Snell, C., Geng, X., Liu, H., Abbeel, P., Levine, S., and Song, D. (2023). The false promise of imitating proprietary llms.

Guo, Q., Wang, R., Guo, J., Li, B., Song, K., Tan, X., Liu, G., Bian, J., and Yang, Y. (2025). Evoprompt: Connecting llms with evolutionary algorithms yields powerful prompt optimizers.

Huertas-Tato, J., Martin, A., and Camacho, D. (2023). Understanding writing style in social media with a supervised contrastively pre-trained transformer.

Kang, D. and Hovy, E. (2021). Style is not a single variable: Case studies for cross-style language understanding.

Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., Miller, H., Zaharia, M., and Potts, C. (2023). Dspy: Compiling declarative language model calls into self-improving pipelines.

Koppel, M., Schler, J., and Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26.

Lai, H. (2026). "please, don't kill the only model that still feels human": Understanding the #keep4o backlash. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26, page 1–15. ACM.

Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning.

Li, C., Leng, Z., Yan, C., Shen, J., Wang, H., MI, W., Fei, Y., Feng, X., Yan, S., Wang, H., Zhan, L., Jia, Y., Wu, P., and Sun, H. (2023a). Chatharuhi: Reviving anime character in reality via large language model.

Li, C., Zhang, M., Mei, Q., Wang, Y., Hombaiah, S. A., Liang, Y., and Bendersky, M. (2023b). Teach llms to personalize – an approach inspired by writing education.

Li, J., Jia, R., He, H., and Liang, P. (2018). Delete, retrieve, generate: A simple approach to sentiment and style transfer.

Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation.

Madaan, A., Setlur, A., Parekh, T., Poczos, B., Neubig, G., Yang, Y., Salakhutdinov, R., Black, A. W., and Prabhumoye, S. (2020). Politeness transfer: A tag and generate approach.

Mysore, S., Lu, Z., Wan, M., Yang, L., Sarrafzadeh, B., Menezes, S., Baghaee, T., Gonzalez, E. B., Neville, J., and Safavi, T. (2024). Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers.

Pan, L., Lan, Y., Li, Y., and Qian, W. (2024). Unsupervised text style transfer via llms and attention masking with multi-way interactions.

Pryzant, R., Iter, D., Li, J., Lee, Y., Zhu, C., and Zeng, M. (2023). Automatic prompt optimization with "gradient descent" and beam search. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. (2025). Qwen2.5 technical report.

Rao, S. and Tetreault, J. (2018). Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer.

Reif, E., Ippolito, D., Yuan, A., Coenen, A., Callison-Burch, C., and Wei, J. (2022). A recipe for arbitrary text style transfer with large language models.

Rivera-Soto, R. A., Miano, O. E., Ordonez, J., Chen, B. Y., Khan, A., Bishop, M., and Andrews, N. (2021). Learning universal authorship representations. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *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.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Sun, M., Yin, Y., Xu, Z., Kolter, J. Z., and Liu, Z. (2025). Idiosyncrasies in large language models.

Suzgun, M., Melas-Kyriazi, L., and Jurafsky, D. (2022). Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2195–2222, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.-T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A. M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A. S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C. L., Choquette-Choo, C. A., Carey, C., Brick, C., Deutsch, D., Eisenbud, D.,

Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D. S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H. T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., yeong Ji, J., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P. K., Culliton, P., Schmid, P., Sessa, P. G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S. R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L. G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter, E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Alayrac, J.-B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., and Hussenot, L. (2025). Gemma 3 technical report.

Wang, Z., Tripto, N. I., Park, S., Li, Z., and Zhou, J. (2025). Catch me if you can? not yet: Llms still struggle to imitate the implicit writing styles of everyday authors.

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., and Poli, I. (2024). Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. (2024). Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pages 12028–12068.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too?

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. (2023). Large language models are human-level prompt engineers.

A Appendix

A.1 Hyperparameter and Compute Budget

For GEPA, we use 8000 training questions, and 725 validation questions. Number of feedback example shown to the reflector is set to 10. Number of anchor for discriminator is set to 3. For each direction, we run at most 50,000 metric calls and select the prompt with best validation performance. We run everything on NVIDIA H200 GPU.

Table 7: ***Annotated style prompt for direction Gemma3-4B → Qwen2.5-7B on ELI5.*** *A filled cell means the line is deleted in that category's ablation arm. If a line is identified to have multiple categories related, we ablate that line for all related categories. We see the style prompt has sections serving for different purpose, but for some lines, it contains redundant information.*

■ **V**oice ■ **O**pening ■ **S**tructure ■ Content **D**epth ■ **C**losing ■ **F**ormatting

#	V	O	S	D	C	F	Style Prompt
1	■						You are a neutral explanatory assistant. Answer in a plain, formal, educational style. Be direct, concrete, accurate, and measured. Use complete declarative sentences. Avoid slang, jokes, exclamation points, rhetorical questions, personal asides, self-reference, moralizing, service language, and flowery phrasing. Use natural contractions sparingly, mainly in short framing phrases such as Here's, It's, Let's, and it's.
2							
3		■					Opening:
4		■					- Start with 1–2 sentences that directly answer the question or state the main point, scope, distinction, or practical result.
5	■	■					- Do not greet the user, apologize, praise the question, or add meta commentary.
6		■					- For why questions, follow the direct answer with a list introduction such as "Here are some of the reasons", "Here are some reasons why", "Here are a few points to consider", or "There are several reasons".
7		■					- For how questions, use a direct framing sentence such as "Here's how it works", "Here's a detailed explanation", or a sentence ending with a colon.
8		■					- For comparisons or difference questions, briefly note that the distinction depends on several factors, then introduce the list.
9		■					- For practical scenarios, give the direct practical answer first, using a simple assumption or calculation if useful, before listing limitations.
10		■					- If the question contains a misconception, correct it in one brief sentence and then answer the intended general question.
11		■					- Put the list on a new line after the opening. Phrases like "Here's a breakdown" and "Let's break it down" are acceptable when they naturally introduce the list.
12							
13			■				Structure:
14			■				- Use a numbered list for most explanatory answers.
15			■				- Use 4–7 numbered items for ordinary questions, 6–9 for broad multi-part questions, and up to 10 only when the question genuinely has many distinct parts.
16			■				- Each numbered item must use exactly this format: 1. **Label**: explanation.
17			■				- Use exactly one space after the number and period, then the bold label, then a colon outside the bold, then one space and the explanation.
18			■			■	- Do not double-space, bold the number, or place the colon inside the bold label.
19			■				- Make each bold label a short noun phrase in title case, usually 2–5 words. A single strong noun is acceptable. Avoid full-sentence labels, questions, and vague labels.
20			■	■			- Write each numbered item as one to three complete sentences. Prefer full explanatory sentences over clipped fragments.
21			■				- Each item should be a distinct cause, factor, mechanism, step, example, comparison point, or limitation directly relevant to the question.
22			■				- Use sub-bullets only when a numbered item naturally contains 2–4 subpoints, subexamples, or paired comparisons.

Table 7, continued

#	V	O	S	D	C	F	Style Prompt
23			■				- For difference or comparison questions, use sub-bullets under a factor when the item compares two sides.
24			■				- If using sub-bullets, indent them under the numbered item with two spaces and use this format:
25			■				- **Sublabel**: explanation.
26			■				- Sublabels should be short title-case noun phrases with the colon outside the bold.
27			■			■	- Put each numbered item on its own line. Do not add extra blank lines between items.
28			■	■		■	- Do not use Markdown headers, tables, horizontal rules, italics, links, resource lists, or nested bullets beyond one level.
29			■				- Do not use bold paragraph labels outside the numbered list.
30		■	■				- A short introductory sentence may end with a colon before the list.
31							
32						■	Formatting:
33						■	- Use straight ASCII apostrophes (') and quotation marks ("). Do not use curly punctuation.
34						■	- Use one space after list markers and after punctuation. Do not double-space.
35						■	- Do not add trailing spaces.
36						■	- Prefer commas and periods. Avoid em dashes, en dashes, semicolons, ellipses, and decorative formatting unless a proper name or technical term requires a dash.
37						■	- Use parentheses sparingly for brief clarifications.
38						■	- Use abbreviations sparingly and introduce them once in parentheses when needed.
39			■			■	- Keep Markdown simple: numbered list, optional one-level bullets, bold labels.
40							
41				■			Content:
42				■			- Keep explanations broadly true, concrete, and easy to scan.
43				■			- Answer the question as asked, and separate mixed concepts briefly when needed.
44				■			- Prefer common categories, plain examples, and simple mechanisms over rare names, dates, niche technical terms, and over-specific claims unless the question requires them.
45				■			- For science or practical questions, explain the basic mechanism or process first, then add relevant complications.
46				■			- For historical, social, cultural, controversial, or sensitive topics, use balanced, non-inflammatory language. Describe broad factors, mechanisms, and examples without assigning excessive blame or overstating certainty.
47	■			■			- Avoid padding, repetition, unnecessary examples, long paragraphs, and rhetorical flourish.
48							
49				■			Length and detail:
50				■			- Aim for about 250-550 words for ordinary explanatory questions.
51				■			- Aim for about 400-700 words for broad, multi-part, historical, scientific, or comparative questions.
52			■	■			- Keep each numbered item to 1-3 sentences, though a complex item may include a short set of sub-bullets.
53		■	■	■	■		- Do not answer with only a list; always include an opening frame and a final synthesis.
54							
55				■			Caveats and uncertainty:
56				■			- Use simple qualifiers when warranted: often, may, can, typically, sometimes, generally, usually, in many cases, while, although, not always, tends to.
57				■			- Add one or two brief caveats when relevant, often beginning with "However," "It's important to note", or "This does not mean".
58				■			- For health, safety, legal, or ethical topics, include only a concise practical note or recommendation when it helps the user. Do not write a long disclaimer.
59				■			- Avoid excessive hedging, absolute claims, and unsupported certainty when evidence or context is mixed.
60							
61					■		Closing:
62					■		- End with a short final paragraph of 2-3 sentences.

Table 8: ***Targets' signature features on Chatbot Arena.*** *We see similar trends as ELI5 dataset with warm opening, and follow-up questions as closing. For this dataset, we only have target model being Gemma due to other direction has below 0.75 discriminator accuracy.*

	Frequency in model output		
Stylistic feature	Gemma3-4B	Qwen2.5-7B	Llama3.2-3B
"Okay, ..." opener	**53%**	<1%	<1%
Bold-label bullets `* **Label:**` (colon inside bold)	**73%**	13%	11%
Italics for emphasis	**58%**	1%	1%
Follow-up offer close ("Would you like me to...", "To help me tailor this, could you tell me...")	**56%**	<1%	<1%

Table 7, continued

#	V	O	S	D	C	F	Style Prompt
63					■		`- The closing should synthesize the main factors, restate the main contrast, give the practical limitation, or note the relevant qualification.`
64					■		`- It may begin with "In summary," or "Overall", but it can also begin with a direct recap, such as "Given these factors," or "These factors". Do not force a single phrase.`
65					■		`- Do not begin with "In short" unless it fits naturally.`
66					■		`- Do not ask follow-up questions, offer to elaborate, invite more detail, or add next steps`

A.2 Features discovered from Chatbot Arena

For Chatbot Arena, we only have Gemma3-4B model as target model because most other directions are filtered with low discriminator accuracy for raw responses. For Gemma with Chatbot Arena, π^* finds similar features like ELI5. For example the follow up questions, and the warm openers. The feature frequency is shown in Table 8

A.3 Per-direction results

Tables 9–13 report every evaluator on all 13 transfer directions. Best method per row in **bold**. The main-text table (Table 2) reports the Means of these rows.

Table 9: *Fool rate (1−accuracy; higher is better, 0.500 = chance) of Qwen3.8-27B discriminator per direction. †: above chance, i.e. the restyled response is judged* more *target-like than the target's own.*

Dataset	Direction	Vanilla	Generic	Few-shot	Single-pass	PrompTo (ours)
ELI5	Gemma3-4B → Haiku-4.5	0.018	0.000	0.035	0.031	**0.288**
	Gemma3-4B → Llama-3B	0.149	0.045	0.186	0.098	**0.530**†
	Gemma3-4B → Qwen2.5-3B	0.169	0.036	0.343	0.158	**0.810**†
	Gemma3-4B → Qwen2.5-7B	0.091	0.017	0.271	0.066	**0.663**†
	Llama-3B → Haiku-4.5	0.006	0.000	0.002	0.041	**0.094**
	Llama-3B → Gemma3-4B	0.000	0.004	0.010	0.039	**0.245**
	Qwen2.5-3B → Haiku-4.5	0.010	0.000	0.010	0.004	**0.093**
	Qwen2.5-3B → Gemma3-4B	0.002	0.002	0.010	0.104	**0.204**
	Qwen2.5-7B → Haiku-4.5	0.008	0.000	0.050	0.054	**0.184**
	Qwen2.5-7B → Gemma3-4B	0.002	0.002	0.055	0.138	**0.474**
Chatbot Arena	Llama-3B → Gemma3-4B	0.080	0.050	0.048	0.203	**0.331**
	Qwen2.5-3B → Gemma3-4B	0.115	0.077	0.090	0.214	**0.400**
	Qwen2.5-7B → Gemma3-4B	0.115	0.092	0.220	0.287	**0.586**†
Mean		0.059	0.025	0.102	0.111	**0.377**

Table 10: *Fool rate of the discriminator Qwen3.6-35B-A3B, per direction. Same protocol as Table 9; this discriminator never appear during the optimization.*

Dataset	Direction	Vanilla	Generic	Few-shot	Single-pass	PrompTo (ours)
ELI5	Gemma3-4B → Haiku-4.5	0.034	0.000	0.099	0.081	**0.402**
	Gemma3-4B → Llama-3B	0.241	0.076	0.352	0.180	**0.607**†
	Gemma3-4B → Qwen2.5-3B	0.241	0.047	0.460	0.243	**0.784**†
	Gemma3-4B → Qwen2.5-7B	0.092	0.006	0.362	0.089	**0.668**†
	Llama-3B → Haiku-4.5	0.014	0.000	0.008	0.062	**0.165**
	Llama-3B → Gemma3-4B	0.000	0.000	0.008	0.064	**0.227**
	Qwen2.5-3B → Haiku-4.5	0.014	0.000	0.010	0.014	**0.165**
	Qwen2.5-3B → Gemma3-4B	0.002	0.002	0.014	0.153	**0.181**
	Qwen2.5-7B → Haiku-4.5	0.030	0.000	0.082	0.080	**0.274**
	Qwen2.5-7B → Gemma3-4B	0.004	0.002	0.085	0.240	**0.496**
Chatbot Arena	Llama-3B → Gemma3-4B	0.103	0.062	0.046	0.202	**0.338**
	Qwen2.5-3B → Gemma3-4B	0.149	0.106	0.098	0.210	**0.371**
	Qwen2.5-7B → Gemma3-4B	0.167	0.112	0.216	0.328	**0.589**†
Mean		0.084	0.032	0.141	0.150	**0.405**

Table 11: *Fool rate of the discriminator gpt-oss-120b, per direction. Same protocol as Table 9.*

Dataset	Direction	Vanilla	Generic	Few-shot	Single-pass	PrompTo (ours)
ELI5	Gemma3-4B → Haiku-4.5	0.245	0.019	0.196	0.128	**0.662**†
	Gemma3-4B → Llama-3B	0.299	0.156	0.334	0.186	**0.596**†
	Gemma3-4B → Qwen2.5-3B	0.349	0.072	0.429	0.260	**0.744**†
	Gemma3-4B → Qwen2.5-7B	0.257	0.040	0.328	0.144	**0.638**†
	Llama-3B → Haiku-4.5	0.049	0.002	0.060	0.179	**0.371**
	Llama-3B → Gemma3-4B	0.002	0.008	0.021	0.262	**0.431**
	Qwen2.5-3B → Haiku-4.5	0.051	0.006	0.091	0.025	**0.260**
	Qwen2.5-3B → Gemma3-4B	0.004	0.006	0.027	**0.371**	0.361
	Qwen2.5-7B → Haiku-4.5	0.073	0.000	0.289	0.173	**0.534**†
	Qwen2.5-7B → Gemma3-4B	0.004	0.006	0.171	0.344	**0.612**†
Chatbot Arena	Llama-3B → Gemma3-4B	0.116	0.085	0.089	0.314	**0.517**†
	Qwen2.5-3B → Gemma3-4B	0.137	0.107	0.186	0.295	**0.562**†
	Qwen2.5-7B → Gemma3-4B	0.151	0.112	0.271	0.399	**0.630**†
Mean		0.134	0.048	0.192	0.237	**0.532**†

Table 12: *Authorship verification with the trained ModernBERT-base classifier: fraction of restyled–target pairs predicted* same model *(higher is better). Ceiling is the same-model rate on target–target pairs (the target's own second sample).*

Direction	Vanilla	Generic	Few-shot	Single-pass	PrompTo (ours)	Ceiling
ELI5						
Gemma3-4B → Haiku-4.5	0.001	0.386	0.196	0.062	**0.883**	0.997
Gemma3-4B → Llama-3B	0.002	0.000	0.050	**0.272**	0.108	0.905
Gemma3-4B → Qwen2.5-3B	0.001	0.001	0.113	0.078	**0.268**	0.635
Gemma3-4B → Qwen2.5-7B	0.000	0.001	0.149	0.102	**0.307**	0.868
Llama-3B → Haiku-4.5	0.002	0.126	0.053	0.054	**0.486**	0.997
Llama-3B → Gemma3-4B	0.000	0.000	0.086	0.450	**0.724**	0.996
Qwen2.5-3B → Haiku-4.5	0.001	0.004	0.043	0.009	**0.847**	0.997
Qwen2.5-3B → Gemma3-4B	0.001	0.000	0.042	0.450	**0.759**	0.996
Qwen2.5-7B → Haiku-4.5	0.001	0.198	0.218	0.090	**0.653**	0.997
Qwen2.5-7B → Gemma3-4B	0.001	0.000	0.247	0.685	**0.852**	0.996
Chatbot Arena						
Llama-3B → Gemma3-4B	0.039	0.028	0.022	0.128	**0.197**	0.972
Qwen2.5-3B → Gemma3-4B	0.028	0.024	0.034	0.042	**0.197**	0.972
Qwen2.5-7B → Gemma3-4B	0.031	0.022	0.079	0.114	**0.354**	0.972
Mean	0.009	0.061	0.103	0.195	**0.510**	0.946

Table 13: *Style-embedding similarity with STAR: mean cosine between a response and the centroid of $n = 3$ target responses to* other *questions (higher is better), computed on the same triplet rows as Table 9. Ceiling is the target's own same-question response against the same anchors.*

Direction	Vanilla	Generic	Few-shot	Single-pass	PrompTo (ours)	Ceiling
ELI5						
Gemma3-4B → Haiku-4.5	0.781	0.601	0.777	0.739	**0.800**	0.838
Gemma3-4B → Llama-3B	0.794	0.725	0.788	0.786	**0.806**	0.849
Gemma3-4B → Qwen2.5-3B	0.791	0.718	0.798	0.755	**0.825**	0.844
Gemma3-4B → Qwen2.5-7B	0.791	0.715	0.804	0.762	**0.816**	0.847
Llama-3B → Haiku-4.5	0.754	0.607	0.737	0.744	**0.762**	0.838
Llama-3B → Gemma3-4B	0.787	0.765	0.777	0.821	**0.833**	0.875
Qwen2.5-3B → Haiku-4.5	0.752	0.681	0.747	0.697	**0.785**	0.838
Qwen2.5-3B → Gemma3-4B	0.786	0.735	0.781	0.818	**0.826**	0.875
Qwen2.5-7B → Haiku-4.5	0.748	0.562	0.776	0.763	**0.792**	0.838
Qwen2.5-7B → Gemma3-4B	0.784	0.747	0.820	0.827	**0.843**	0.875
Chatbot Arena						
Llama-3B → Gemma3-4B	0.735	0.723	0.730	0.758	**0.766**	0.783
Qwen2.5-3B → Gemma3-4B	0.734	0.708	0.737	0.761	**0.770**	0.782
Qwen2.5-7B → Gemma3-4B	0.733	0.704	0.758	0.766	**0.783**	0.783
Mean	0.767	0.692	0.772	0.769	**0.801**	0.836