WHAT IS IT LIKE TO WRITE LIKE HAIKU?

Anonymous authors
Paper under double-blind review

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), because what they missed most was the "warm" tone with which GPT-4o wrote. While GPT-4o may in fact write more warmly than GPT-5, such a description is not enough to fully articulate the style. Simply telling GPT-5 to write more warmly would not make the experience 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 model that has never seen the original can reproduce the 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 producing a particular model's style.

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.

We introduce **PrompTo**, a framework that searches for a complete description of a target model and tests it by reproduction. The description is in the format of a system prompt, which we call a style prompt. To test the style prompt, we use a source model to generate the restyled text, and a discriminator is asked to assess the likeness between the text generated by target model and restyled source model output. We report how much of the gap between raw source response and target-like text a description closes as Style Gap Recovered (SGR). SGR of 0% means the style prompt does nothing helpful for the transformation, 100% means the source model is indistinguishable from the target because of the style prompt. We use a discriminator with a reflector to search for a complete style prompt. The discriminator's reasoning is passed to a reflection model that revises the style prompt, as shown in Figure 1. In practice, we consider three different kinds of evaluators: an LLM-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

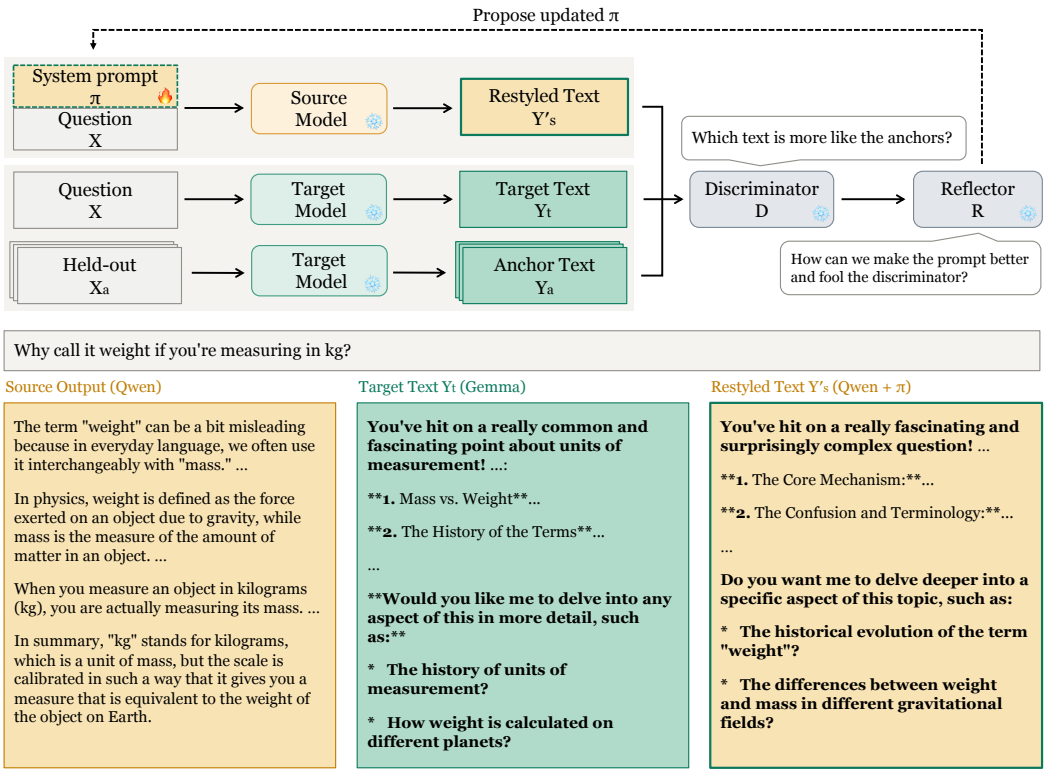

Figure 1: **The PrompTo framework for stylistic transfer.** *Top:* the source model answers X under a trainable style prompt π to generate the restyled text Y_s'. The target model answers X and held-out X_a to generate Y_t and Y_a. Discriminator D picks which of Y_s', Y_t is closer in style to Y_a. The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

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. Specifically, we find actionable mechanisms behind the styles. 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 but no AI-like phrases. Style prompts optimized with one source model and target model can be used to transfer style for other source models to the same target model with most of the SGR retained. This indicates that style prompts mainly describe the target's style rather than suppress the traits in source model they were optimized with. The style prompts that PrompTo generates have a similar structure and different models have different features that make them distinguishable. 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 PrompTo for Stylistic Transfer

PrompTo searches for a style description in the form of a system prompt called style prompt that makes a source model M_s write like a target model M_t (Figure 1). Each iteration does three things in the following order: generating restyled responses under π; evaluating π with an LLM discriminator; proposing updated π with a reflection model using the discriminator's reasoning and failure cases.

2.1 DISCRIMINATOR

The discriminator D evaluates the style prompt. 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. We let Y_a answer different questions from Y_t to avoid content shortcuts. Otherwise, D could select Y_t by comparing content alone. 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: the probability that D predicts Y_s' as being more similar to Y_a. A fool rate of 50% means the restyled responses are almost identical in terms of style to the target responses. The search for such a style prompt can be written as:

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

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 mostly has a fool rate of nearly zero. Any prompt optimizer can be used here. For PrompTo, we use GEPA (Agrawal et al., 2026), because it takes in natural language feedback. This feedback makes the optimization signal rich. GEPA is a reflective prompt evolution algorithm. It maintains a pool of candidate prompts and iteratively mutates them with a reflection model. At each step PrompTo does the following things: generating a batch of restyled responses Y_s' with the source model M_s under the current style prompt π; evaluating π by asking an LLM discriminator to predict which of Y_s' and Y_t is generated with M_t; and having an LLM reflector R see the response pairs that D correctly judged along with D's reasoning. It then proposes new π. Only the style prompt changes, with M_s, M_t, D, and R stays frozen. Therefore proprietary model can serve as any role.

2.3 EVALUATION

Evaluating on a single LLM discriminator could cause overfitting because it is used for the GEPA optimization. We therefore evaluate π^* on other 2 evaluators that are not LLMs, and 2 LLM discriminators that never appear during the optimization.

Discriminators. We report the fool rates of the Qwen3.8-27B discriminator, which is used during GEPA optimization, and two discriminators that never appear appear during the optimization (Qwen3.6-35B-A3B and gpt-oss-120b). Each discriminator is evaluated with the same protocol described in Section2.1. 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). A ModernBERT-base encoder (Warner et al., 2024) receives two responses and predicts whether the same model generate them. Training pairs are built from raw, unprompted responses of all five models on the training split, and balanced between same-model and different-model pairs. At test time we present restyled–target pairs on the same question and report the AV ratio which is the fraction of them being predicted as same-model.

Style embeddings. We use STAR (Huertas-Tato et al., 2023), a contrastively trained style encoder, in the same setup as LLM discriminator. For each test question we embed n anchor responses from M_t on other questions and average them into a target centroid, then compute the cosine similarity of the restyled response to the centroid. We use the same anchor setup to avoid content shortcut.

Style gap recovered. The 13 transfer directions vary in difficulty, 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:

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

where $F(\cdot)$ is the fool rate of a single 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 such that it makes a source model's responses indistinguishable from a target's. We compare them against other kinds of description with the same test. Our experiments span 13 direction-dataset configurations built from 5 models and 2 question datasets, as shown in Table 3.

Datasets. We evaluate our method on two datasets of open-ended questions, because open-ended responses leave room for stylistic variation to emerge. The two datasets are ELI5 (Fan et al., 2019), explanations of everyday questions from the r/explainlikeimfive subreddit, and Chatbot Arena Conversations (Zheng et al., 2023), 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. We select 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) as source and target models. These four open-weight models serve as both sources and targets, while Claude Haiku 4.5 serves only as a target for budget reasons. The reflection model and discriminator model are both Qwen3.8-27B for its thoughtful reflection and accurate judgment. 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 signal for training would be compromised. 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.

Baselines. We have four baselines for comparison. Each baseline is a different kind of description of the target's style. We don't include VibeCheck (Dunlap et al., 2025) as a baseline because it does not give any actionable description of transferring the style of one model to another.

- **Vanilla** gives the source model no description to follow. Its score is the floor of SGR and measures how distinguishable the two models are out of the box.
- **Generic** gives only the target's name (e.g. "Write in the style of Qwen"). It tests whether the source model already contains the information about the target model from pretraining.
- **Few-shot** (Wang et al., 2025) replaces the description with three target responses answering other questions. It is an implicit description because the style is in the examples but never stated explicitly. This tests whether the source model can learn from the examples, and follow the style consistently.
- **Single-pass** gives the reflection model the same 10 target responses that PrompTo starts from, and asks it to generate the style description with no revision. This tests the contribution of discriminator-driven prompt optimization.

3.2 MAIN RESULTS

Table 1 shows the averaged results over the 13 directions listed in Table 3. PrompTo's optimized style prompts close the source-target gap more than any other kind of description. With LLM discriminators, the style prompt recovers 74.8% of the source–target gap on average. Few-shot, as the strongest baseline, only recovers 15.2% of the gap. The two non-LLM evaluators agree with the LLM discriminators. STAR and AV classifier also rank PrompTo's style prompt as the best. This indicates the style prompts capture properties visible to different kinds of evaluators. Among the

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**

Table 1: **Main results of PrompTo**, averaged over the 13 directions across three kinds of evaluators. Fool rate and its Style Gap Recovered (SGR) are averaged across three LLM discriminators; AV ratio is the AV classifier's same-model rate; Cos. sim. is the STAR embedding cosine similarity between the text and centroids. Higher is better throughout.

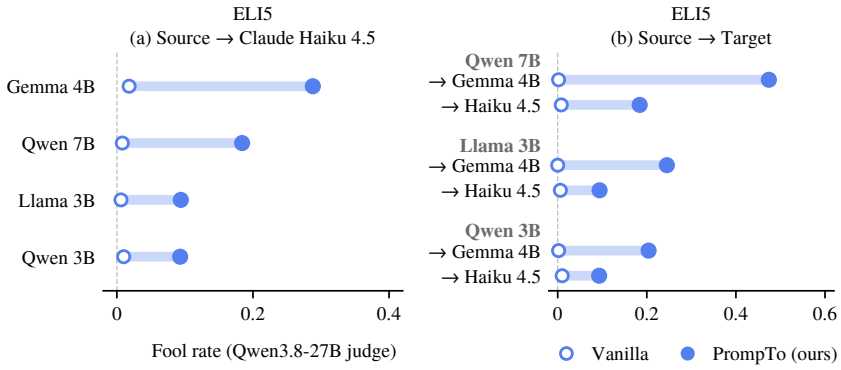

Figure 2: **Effect of source and target model.** *(a)* Target fixed to Claude Haiku 4.5, source varied. *(b)* For each source, target varied between Gemma3-4B and Claude Haiku 4.5. This shows final fool rate depends on the source model capacity and target difficulty.

baselines, Generic is the worst with average fool rate being 3.5%. Telling a source model to "write like Qwen" makes the output less Qwen-like, because the source model does not have knowledge of the style of other models. Few-shot and Single-pass both have modest SGR. Few-shot is stronger on style embeddings, and Single-pass is stronger on the AV classifier. In summary, the rows of Table 1 shows how much of style information does each kind of description carry. A model name alone carries no style property. Examples or a simple description without revision carry a little style information. Only PrompTo's style prompt, which is optimized with a discriminator, carries most of it.

3.3 Variation across configurations.

The average in Table 1 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 shows how source model or target model affect the fool rate. Evaluation results for all direction separately are shown in SectionA.3

Source model. With the target fixed to Claude Haiku 4.5 on ELI5 (Figure 2a), the four sources start from nearly identical raw fool rates but diverge once π^* are applied. Gemma3-4B is the highest, reaching 28.8%, Qwen2.5-7B being in the middle, and the two 3B models being the two lowest. The ordering shows executor's capacity difference as Qwen2.5-7B beats Qwen2.5-3B on every target they share, while Llama3.2-3B and Qwen2.5-3B are having similar performance. The gap is mainly due to how well each source model follows the style prompts. The style prompts targeting Haiku specify the same feature regardless of source, but Llama3.2-3B and Qwen2.5-3B follows them less reliably. Weaker models also misread instructions. Qwen2.5-3B's style prompt shows "Main reason" and "Practical takeaway" as examples of possible section labels, but the model treats 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. For every source model that target both Gemma and Haiku in (Figure 2b), Gemma's style is easier to describe and execute than Haiku. The gap is largest for Qwen2.5-7B with its 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 their fool rates targeting Gemma being roughly two times the fool rates targeting Haiku. The opening of the 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. Because of that, it's harder for a smaller model to learn this new way of writing in a content-dependent manner. 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, PrompTo drops quality by 0.45 points, roughly half the drop of the Generic and Few-shot baselines (0.94 and 0.90 points, respectively), which also transfer far less style. Single-pass drops less (0.31 points) but also transfers less style (Table 1).

To alleviate the quality drop, we design a quality-aware variant of PrompTo. In the original PrompTo, 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 GEPA objective.

As shown in Figure 5, the quality-aware variant roughly halves the quality drop, making it go from 0.45 to 0.23 points. The trade-off is a lower fool rate, from 42.5% to 33.5% averaged over the three discriminators. 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; your second job is to present the answer in a scannable, upbeat, structured way." Such 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 is what brings up quality but lost some fool rate.

4 Style Prompt Study

4.1 Model Signature

Studying the optimized style prompt with the following model being the target, we find the mechanism behind why these models (Qwen2.5-7B, Gemma3-4B, and Claude Haiku 4.5) each has a namable persona. PrompTo's style prompts discover Gemma as an enthusiastic tutor, because it uses warm openings and asks follow-up questions; Qwen as a neutral assistant, because it goes directly to the question and does nothing else; and Haiku as a knowledgeable friend, because it gets straight to the answer with warm voice but no AI-like phrases. We do not include Llama3.2-3B here because we only have one direction with it being the target model, and its style is similar to Qwen's. For each model, we measure the frequency of their distinctive features with regex. All of the features discussed below appear in their style prompts shown in Figure 3.

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 π^* recover is the closing phrases. "Delve deeper" appears in 87% of Gemma's ELI5 responses and almost never in other models. Warm openers together appear in 80% of Gemma's response across both dataset and also almost never in other models. According to the discriminator's reasoning, these two sets of phrases are what distinguish Gemma from the other models.

Stylistic feature	Frequency in model output		
	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**

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

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 targeting Qwen. The style prompts also spend more effort on what to avoid.

Qwen's signature is largely absent. It never asks a question back, provides links, or write disclaimers. Its most common opening phrase only covers 0.2% of its responses while Gemma's covers 27.5%. The unique feature is the body format. Most of Qwen2.5-7B's responses are a single flat numbered list. The fact that Qwen's responses are plain makes Qwen reaches the highest SGR among the three targets. When Gemma is used as source model targeting Qwen, the transfer is largely a subtraction. It only needs to drop the opening and closing phrases and write the body as a numbered list. 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 not to 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. Most of its responses open with a Markdown heading, which no other models do, and every π^* catches it. Each style prompt even specifies how to write one. They cover how specific the title should be, what to avoid, and often a few examples. π^* describes the structure of the response as a short document.

Haiku is hard because of the capability difference. Once the style matches, the only remaining part to spot the difference is the answer depth and correctness. No 3–7B model can close this gap with a single system prompt. gpt-oss-120b's judgement focus on surface level styles, therefore it is often fooled when targeting Haiku. However, Qwen3.8-27B as the most capable LLM discriminator makes the right decision more often.

4.2 Style Prompt Generalization

To find whether π^* describes the target's style or the difference in style between a particular source and the target, we let the source model swap the style prompts. For one style prompt that transfers the style of a source model to the style of a target model, we call that source model the owner of the style prompt because the style prompt is optimized for that source model specifically. Any other source models running the style prompt are the transferee. If π^* mainly describes the target's style, we should see that the transferee can still achieve target-like style despite running style prompt op-

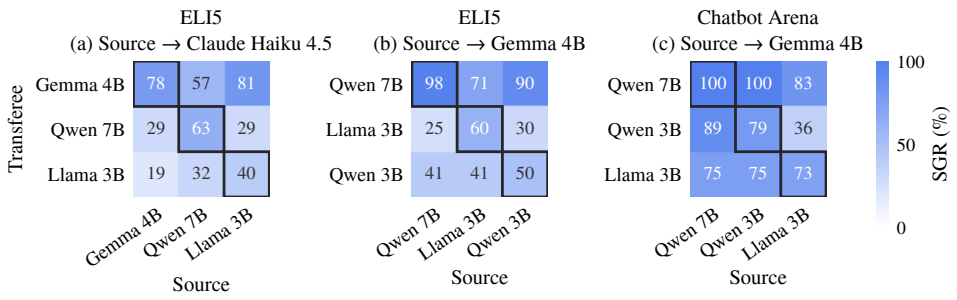

Figure 4: **Style prompts transfer across source models.** Each panel is one target on one dataset. Each row is a source model executing a style prompt and each column is the source model the prompt was optimized for. Boxed diagonal cells are each model running its own prompt and off-diagonal cells are transferred prompts.

		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

Table 2: **SGR of different ablations across categories in π^*.** We do ablation in two directions, letting Qwen and Gemma model be source or target model. We see 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 body structure.

timized for someone else. Figure 4 supports it with transferee retaining 78% of the SGR comparing to owner on average. In Figure 4 (a) Gemma3-4B on any transferred prompt beats Llama3.2-3B on its own. This shows the source model's capability is contributing to the SGR here, and swapping style prompt does not hurt that much comparing to swapping to a weaker source model.

The text also agrees that each π^* mainly describes the target style. We use Qwen-3-Embedding-8B to turn each π^* into embeddings. We compare cosine distances between three groups of pairs: different source with same target; same source with different target; and different source with different target. On average any unrelated pairs has distance of 0.342. Style prompt from the same source for different target is slightly closer with 0.338, and style prompts that describe the same target from different sources are much closer with 0.211. Sharing a target pulls two style prompts together but sharing a source does not. This again supports π^* describes the target's style rather than depending on the difference between the source style and target style.

4.3 Ablation

To see which parts of the π^* actually carry the target style, we ablate 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 effect of the ablation. We manually partition each π^* into six categories shown in Table 2. Both style prompts share the same structure, but spends more effort on different things. The Gemma-target style prompt spends more on voice and closing, while the Qwen-target style prompt spends more on body structure. For the lines that shares two categories, we ablate it for both ablation runs to ensure no leakage. The way we categorize is shown in Table 4.

Table 2 shows feature importance is target-specific. For Gemma3-4B, removing the opening drops SGR from 95% to 15%, removing the closing drops it to 21%, and removing voice drops it to 34%. On the other hand, dropping content depth or structure costs less SGR. For Qwen2.5-7B, body structure becomes the most important category ($100\% \rightarrow 45\%$), 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 in Section 4.1 with frequency finding. In addition, not everything in the style prompt contributes to SGR positively. Formatting is described in both style prompts, but

removing it barely moves SGR. All of these rankings are stable across the three LLM discriminator with $\rho \geq 0.94$ within each direction.

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 & 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 & 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); and fine-tuning approaches adapt model weights on the author's corpus (Zhang et al., 2018).

5.2 Authorship Analysis

Authorship analysis asks who wrote a text. Its two formulations are attribution, which selects an author from a closed candidate set, and 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 & 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 the most recent work, it adds 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 the 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.

AI USE STATEMENT

In this work, we used generative AI tools for writing polishing and idea execution. Specifically, we used generative AI tools for paper proofreading, grammar checking, and coding. We have not used generative AI tools for other tasks with required disclosure. For tasks with recommended disclosure, we used generative AI tools for polishing figures and tables aesthetically. We have reviewed all AI-assisted work. We checked and tested all generated code. We manually reviewed all edit suggestions for text and figures generated by generative AI tools before incorporating them. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our study involves no human subjects; all inputs are questions from the public ELI5 and Chatbot Arena Conversations datasets, and all responses are generated by the models we study. While PrompTo can describe a model's writing style in interpretable natural language and let a source model reproduce the style, it can also be misused, for example to pass off one model as another or to evade style-based model attribution. We therefore encourage services that restyle a model to disclose which model serves the user, and attribution tools to rely on signals beyond surface style.

REPRODUCIBILITY STATEMENT

We will open-source the code for learning the style prompts, and for all evaluation protocols. We have submitted related code and data as supplemental material. All data processing procedures and sampling parameters are included. Optimized style prompts are also included in supplemental material, therefore result can be inspected without re-running optimization. Hyper-parameters related to optimization is included in AppendixA.1.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. Gepa: Reflective prompt evolution can outperform reinforcement learning, 2026. URL https://arxiv.org/abs/2507.19457.

Anthropic. System card: Claude Haiku 4.5, October 2025. URL https://www.anthropic.com/claude-haiku-4-5-system-card.

Avanti Bhandarkar, Ronald Wilson, Anushka Swarup, and Damon Woodard. Emulating author style: A feasibility study of prompt-enabled text stylization with off-the-shelf LLMs. In Ameet Deshpande, EunJeong Hwang, Vishvak Murahari, Joon Sung Park, Diyi Yang, Ashish Sabharwal, Karthik Narasimhan, and Ashwin Kalyan (eds.), *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pp. 76–82, St. Julians, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.personalize-1.6. URL https://aclanthology.org/2024.personalize-1.6/.

Benedikt Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. Explainable authorship verification in social media via attention-based similarity learning, 2019. URL https://arxiv.org/abs/1910.08144.

John F. Burrows. 'delta': a measure of stylistic difference and a guide to likely authorship. *Lit. Linguistic Comput.*, 17:267–287, 2002. URL https://api.semanticscholar.org/CorpusID:20390195.

Lisa Dunlap, Krishna Mandal, Trevor Darrell, Jacob Steinhardt, and Joseph E Gonzalez. Vibecheck: Discover and quantify qualitative differences in large language models, 2025. URL https://arxiv.org/abs/2410.12851.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering, 2019. URL https://arxiv.org/abs/1907.09190.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia

Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms, 2023. URL https://arxiv.org/abs/2305.15717.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Evoprompt: Connecting llms with evolutionary algorithms yields powerful prompt optimizers, 2025. URL https://arxiv.org/abs/2309.08532.

Javier Huertas-Tato, Alejandro Martin, and David Camacho. Understanding writing style in social media with a supervised contrastively pre-trained transformer, 2023. URL https://arxiv.org/abs/2310.11081.

Dongyeop Kang and Eduard Hovy. Style is not a single variable: Case studies for cross-style language understanding, 2021. URL https://arxiv.org/abs/1911.03663.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL https://arxiv.org/abs/2310.03714.

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

2009. doi: https://doi.org/10.1002/asi.20961. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20961.

Huiqian Lai. "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, pp. 1–15. ACM, April 2026. doi: 10.1145/3772318.3791351. URL http://dx.doi.org/10.1145/3772318.3791351.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/abs/2104.08691.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. Chatharuhi: Reviving anime character in reality via large language model, 2023a. URL https://arxiv.org/abs/2308.09597.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. Teach llms to personalize – an approach inspired by writing education, 2023b. URL https://arxiv.org/abs/2308.07968.

Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer, 2018. URL https://arxiv.org/abs/1804.06437.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021. URL https://arxiv.org/abs/2101.00190.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach, 2020. URL https://arxiv.org/abs/2004.14257.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers, 2024. URL https://arxiv.org/abs/2311.09180.

Lei Pan, Yunshi Lan, Yang Li, and Weining Qian. Unsupervised text style transfer via llms and attention masking with multi-way interactions, 2024. URL https://arxiv.org/abs/2402.13647.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.494. URL https://aclanthology.org/2023.emnlp-main.494/.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Sudha Rao and Joel Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer, 2018. URL https://arxiv.org/abs/1803.06535.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. A recipe for arbitrary text style transfer with large language models, 2022. URL https://arxiv.org/abs/2109.03910.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. Learning universal authorship representations. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 913–919, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.70. URL https://aclanthology.org/2021.emnlp-main.70/.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009. doi: https://doi.org/10.1002/asi.21001. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001.

Mingjie Sun, Yida Yin, Zhiqiu Xu, J. Zico Kolter, and Zhuang Liu. Idiosyncrasies in large language models, 2025. URL https://arxiv.org/abs/2502.12150.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2195–2222, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.141. URL https://aclanthology.org/2022.emnlp-main.141/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Zhengxiang Wang, Nafis Irtiza Tripto, Solha Park, Zhenzhen Li, and Jiawei Zhou. Catch me if you can? not yet: Llms still struggle to imitate the implicit writing styles of everyday authors, 2025. URL https://arxiv.org/abs/2509.14543.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL https://arxiv.org/abs/2412.13663.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too?, 2018. URL https://arxiv.org/abs/1801.07243.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers, 2023. URL https://arxiv.org/abs/2211.01910.

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 3: Source-target directions covered in our evaluation.

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 4: **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.

Voice **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.

Table 4, continued

#	V	O	S	D	C	F	Style Prompt
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.
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".

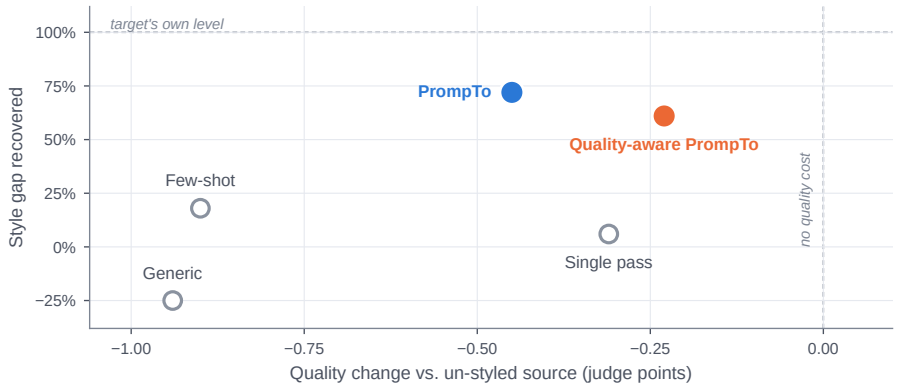

Figure 5: **Quality vs. style.** Y-axis represents SGR, higher the better; and X-axis represents how much of quality is lost on a 10 point scale. We see Quality-aware PromptTo recovers the quality drop caused by PromptTo, but recovers less style gap. Both method only cost less than 0.5 points in a 10 point scale evaluated with Qwen3.6-35B-A3B. We use 10 directions with ELI5 dataset.

Table 4, continued

#	V	O	S	D	C	F	Style Prompt
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.
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 5

A.3 Per-direction results

Tables 6–10 report every evaluator on all 13 transfer directions. Best method per row in **bold**. The main-text table (Table 1) reports the Means of these rows.

Stylistic feature	Frequency in model output		
	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 5: **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.

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 6: 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	0.034	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 7: Fool rate of the discriminator Qwen3.6-35B-A3B, per direction. Same protocol as Table 6; this discriminator never appear during the optimization.

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 8: Fool rate of the discriminator gpt-oss-120b, per direction. Same protocol as Table 6.

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 9: 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	0.781	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

Table 10: 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 6. Ceiling is the target's own same-question response against the same anchors.