Tuning-Free Personalized Alignment via Trial-Error-Explain In-Context Learning

Hyundong Cho[1]*, **Karishma Sharma[2], Nicolaas Jedema[2], Leonardo F. R. Ribeiro[2],**
Alessandro Moschitti[2], Ravi Krishnan[2], Jonathan May[1]
[1]University of Southern California, Information Sciences Institute [2]Amazon AGI
hd.justincho@gmail.com

Abstract

Language models are aligned to the collective voice of many, resulting in generic outputs that do not align with specific users' styles. In this work, we present *Trial-Error-Explain In-Context Learning* (**TICL**), a tuning-free method that personalizes language models for text generation tasks with fewer than 10 examples per user. TICL iteratively expands an in-context learning prompt via a *trial-error-explain* process, adding model-generated negative samples and explanations that provide fine-grained guidance towards a specific user's style. TICL achieves favorable win rates on pairwise comparisons with LLM-as-a-judge up to 91.5% against the previous state-of-the-art and outperforms competitive tuning-free baselines for personalized alignment tasks of writing emails, essays and news articles. Both lexical and qualitative analyses show that the negative samples and explanations enable language models to learn stylistic context more effectively and overcome the bias towards structural and formal phrases observed in their zero-shot outputs. By front-loading inference compute to create a user-specific in-context learning prompt that does not require extra generation steps at test time, TICL presents a novel yet simple approach for personalized alignment.

1 Introduction

Large language models (LLMs) are biased towards generic outputs as they are trained to align to an aggregate preference to be generally useful (Padmakumar and He, 2024; Rafailov et al., 2024; Lee et al., 2023; Wang et al., 2023). Yet end users often have specific needs that call for personalized text generation, such as writing an email (Kumar et al., 2024; Salemi et al., 2023). Hence, adapting LLMs for personalized text generation has drawn a growing interest (Jang et al., 2023; Shaikh et al., 2024; Mysore et al., 2023; Li et al., 2024), but

Figure 1: *Top:* The author's text contains rhetorical questions and more colloquial and informal phrases. *Middle*: Few-shot in-context learning is insufficient for adapting to an author's style and model biases for formal phrases persist. *Bottom:* TICL helps overcome model biases and applies the author's style more consistently. Model outputs are with gpt-4o-2024-0806

prior work on personalized text generation either rely on large amounts of personal data (Li et al., 2024, 2023; Mysore et al., 2023) or parameter updates that result in cumbersome per-user[1] model parameters (Shaikh et al., 2024; Liu et al., 2023).

These are critical limitations. Personalization is impractical when premised on users providing ample data because most are not willing to exert such effort (Tétard and Collan, 2009) or are concerned

---

* Work was done while HC was an intern at Amazon AGI.

[1]We use the term "user" and "author" interchangeably throughout this work. A user is anyone who provides personally written text that style can be learned from.

about privacy (Plant et al., 2022). In addition, fine-tuning may not even be a viable option. The best performing LLMs are most often only accessible via APIs or too large that fine-tuning them is prohibitive even if they are open-sourced (Touvron et al., 2023; Firsich and Rios, 2024; Anthropic, 2024). Can we develop a method that overcomes these limitations such that it is effective with small amounts of data and requires no parameter updates?

To address this problem, we propose **T**rial-**E**rror-**E**xplain **I**n-**C**ontext **L**earning (**TICL**), a *tuning-free* adaptation of Trial-and-Error Fine-tuning (TEFT) (Shaikh et al., 2024; Gulcehre et al., 2023; Song et al., 2024). The key idea of TEFT is to iteratively generate synthetic negative samples to create a preference dataset and apply a preference optimization method such as DPO (Rafailov et al., 2024) to learn a more nuanced preference even with a small amount of initial training data. TICL takes a similar approach but only relies on scaling inference compute to expand an in-context learning (ICL) prompt to include prior model-generated negative samples (*trial-error*) and corresponding explanations that describe their shortcomings (*explain*) to better understand a user's personal style.

We evaluate the effectiveness of TICL with GPT-4o and Claude 3 Sonnet on two personalized writing datasets consisting of emails, essays, and news articles. Using an extensively verified LLM-as-a-judge setup as the evaluator for comparing style similarity between two pairs of generated text with a user's text, we find that TICL achieves higher win rates (i.e. more frequently considered more similar in pairwise comparisons) than all other competitive prompting-based baselines, including Chain-of-Thought (Wei et al., 2022) and OPRO (Yang et al., 2024) for models. TICL also achieves an average win rate of 76.6% against the previous state-of-the-art (Shaikh et al., 2024). We ablate TICL extensively and find that, while every procedure in TICL contributes to stronger performance, explanations are crucial, as it is responsible for up to 77% of the gains relative to a simple ICL baseline.

Lastly, we also share lexical analysis that shed light on the performance gaps between various models and methods. As illustrated in Figure 1, compared to ICL, we find TICL more reliably reduces LLM bias towards structural and formal phrases, e.g., "*additionally*" and "*therefore*", and instead use more colloquial ones, e.g., "*so why*" and "*honestly.*" We also discover that Claude 3 Sonnet's consistent dominance over GPT-4o is attributed to Claude 3 Sonnet more reliably generating phrases that elicit opinions, such as "*believe*" and "*feel that,*" which are frequently seen in the users' texts.

2 Methodology

Trial-Error-Explain In-Context Learning (TICL) is a method for expanding an ICL prompt through tuning-free correspondences of the three main stages in Trial-and-Error Fine-tuning (TEFT). Therefore, we describe TICL by going through each main stages of TEFT and the corresponding procedure in TICL, as illustrated in Figure 2. TEFT refers to a method that (*i*) fine-tunes a model with supervised fine-tuning (SFT) on a small set of training data (behavior cloning), (*ii*) uses the trained model to generate outputs that are considered less preferred to those from the training data (exploration), (*iii*) and fine-tunes the model further with a preference optimization method, such as DPO (learning). This method has been applied to various tasks in previous work with minor differences, such as machine translation with ReST (Gulcehre et al., 2023), personalized text generation with DITTO (Shaikh et al., 2024), and trajectory planning with ETO (Song et al., 2024).

In TICL, we first replace SFT with ICL for behavior cloning (§**??**). Exploration is mostly the same except that TICL updates the prompt at the step level while TEFT updates the policy at the epoch level in order to avoid augmenting the prompt with redundant errors (§2.2). Lastly, for learning, we substitute preference optimization with prompt augmentation (§2.3). TICL generates explanations that analyzes the differences between the generated text and the user's actual text. Then, the generated text is labeled as a negative sample and the explanation are added to the prompt. We elaborate on each correspondence and share additional details in the following sections.

2.1 Behavioral cloning

As shown in the first step in Figure 2, TEFT uses supervised fine-tuning (SFT) to fine-tune a model π_θ roughly towards the desired behavior with the initial training data. This helps with generating more relevant negative samples in the exploration phase (Song et al., 2024).

Formally, with the initial training data $\mathcal{D}^\mathcal{T} = \{(x_i, y_i)|_{i=1}^N\}$, where x is the task and y is the user's text, SFT uses the negative log likelihood
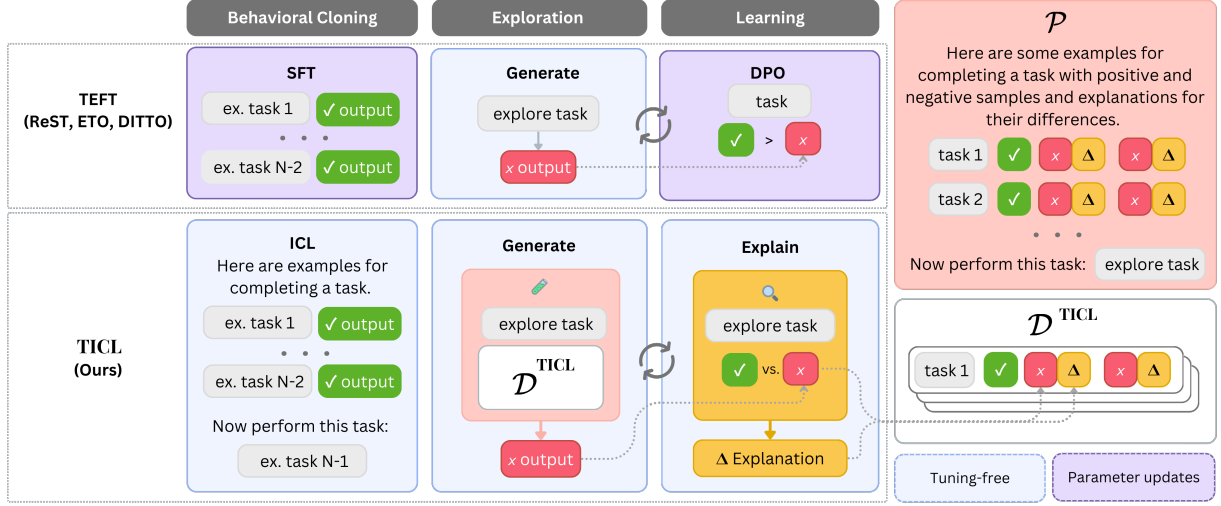
Figure 2: TICL methodology overview. Instead of SFT, TICL starts with a few-shot ICL for behavior cloning. Then, TICL repeatedly generates an output for a task and a corresponding explanation that critiques the stylistic difference between the output and the user's text. If the output is considered not stylistically consistent, it and its explanation are added to the prompt.

(NLL) loss to learn from $\mathcal{D}^{\mathcal{T}}$ to fine-tune a pre-trained LLM π_θ parameterized by θ such that:

$$\mathcal{L}_{NLL}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathcal{T}}} \sum_{t=1}^{T} \log \pi_\theta(y_t|y_{1:t-1}, x)$$

SFT → Few-shot ICL. To substitute SFT, we use $\mathcal{D}^{\mathcal{T}}$ to form an in-context learning (ICL) prompt and perform few-shot ICL (Brown et al., 2020). The resulting ICL prompt is defined as:

$$\mathcal{P}\left(x_i, \mathcal{D}^{\texttt{icl}}\right), \text{where}$$
$$\mathcal{D}^{\texttt{icl}} = \{(x_k, y_k, S_k)\}|_{k=1}^{K}, S = \{(\tilde{y}_l, e_l)\}|_{l=1}^{L} \quad (1)$$

Each sample in $\mathcal{D}^{\texttt{icl}}$ is a tuple of task x, user's text y, and a set of model-generated outputs and their corresponding explanations $S = \{(\tilde{y}_l, e_l)\}|_{l=1}^{L}$ from previous iterations L. $\mathcal{P}$ is a templating function that takes $\mathcal{D}^{\texttt{icl}}$, and the target input x_i and forms an ICL prompt.[2] In the first step of TICL, there are no previous model-generated outputs and explanations, so Eq.1 is simply $\mathcal{P}\left(x_i, \{(x_k, y_k)|_{k=1}^{K}\}\right)$.

2.2 Exploration

The objective of the exploration phase, the middle stage in Figure 2, is to generate negative samples that will be used for learning. TEFT and TICL is mostly similar except for minor technical differences.

In TEFT, the model π_θ resulting from SFT or the previous iteration of DPO generates output $\tilde{y}$ for the target task x without any prompt templating. In contrast, TICL takes prompts resulting from Eq.1 as input:

$$\tilde{y}_i \sim \pi_\theta\left(p\left(x_i, \mathcal{D}^{\texttt{icl}}\right)\right) \quad (2)$$

Another difference is that TEFT generates outputs for the full set of inputs in $\mathcal{D}^{\mathcal{T}}$: $\{\tilde{y}_i \sim \pi_\theta(x_i)|x_i \in \mathcal{D}_0^{\mathcal{T}}\}$, prior to the learning stage. TICL only generates one output from a single sample before moving on to the learning phase – think of batch size $= 1$. We observe that generating outputs for all samples in $\mathcal{D}^{\mathcal{T}}$ leads to multiple $\tilde{y}$ that contain redundant modes of failure. We empirically see that this inefficiency can be avoided by exploring a single $\tilde{y}$ at a time.

2.3 Learning

In the learning phase for TEFT, the preference optimization samples $\mathcal{D}_i^{\texttt{pref}}$ collected from the exploration phase are used for fine-tuning π_θ with the use of a reference model π_{ref}, usually a pre-trained language model for preventing overfitting and reward hacking, and a preference optimization method, such as direct preference optimization (DPO) (Rafailov et al., 2024):

Preference optimization → Prompt augmentation with negative samples and explanations The main benefit of preference optimization is that

---

[2] We share the details of $\mathcal{P}$ and all other prompts used in this work in Appendix F.

the fine-tuned model learns a fine-grained understanding of the differences between reference outputs and negative samples. With TICL, we provide similar information in context via the negative samples and their corresponding explanations that analyze the difference between them and the reference outputs.

We first let the model analyze the differences between the reference output y and its respective negative output $\tilde{y}$. This step is akin to rationale generation (Wang et al., 2022; Wei et al., 2022; Zhou et al., 2022), but the difference with prior work is that TICL's rationale focuses on explicitly extracting the differences between the reference output and the negative output, rather than producing a rationale as an intermediate step before fulfilling the target task.

For this step, we generate explanations using an explanation generation prompt template $\mathcal{E}$ that takes in y and $\tilde{y}$:

$$(e_i, v_i) \sim f(\mathcal{E}(y_i, \tilde{y}_i)) \qquad (3)$$

where e_i is the explanation and v_i is a validator Boolean that checks whether $\tilde{y}_i$ is comparable in quality or accuracy as y_i. $f(\cdot)$ can be any function or combination of functions that generates e_i and v_i. We use the same π_θ unless specified otherwise and have $\mathcal{E}$ instruct the model to generate both e_i and v_i in a JSON format. If $v_i \neq$ True, then we add $(\tilde{y}_i, e_i)$ to S_i: (x_i, y_i)'s set of model-generated outputs and explanations. Otherwise, we prevent good $\tilde{y}$ from being included in the following iterations as negative samples, in order to avoid noise and the context becoming unnecessarily longer.

Iterations and checkpointing In summary, TICL starts with standard ICL to substitute SFT and repeats (*i*) exploration with an expanding dataset to generate a candidate negative sample (Eq.2) and (*ii*) learning to generate an explanation, expand the dataset with the negative sample and explanation if the sample is deemed a proper negative sample (Eq.3). A formal algorithm for TICL is defined in Algorithm 1. Denoting the augmented dataset that starts from $\mathcal{D}^{\mathcal{T}}$ as $\mathcal{D}^{\text{TICL}}$, we keep track of the version of $\mathcal{D}^{\text{TICL}}$ that attains the best performance on the validation set, measuring it every n steps.

Algorithm 1: TICL algorithm

Input: π_θ: LLM, $\mathcal{D}^{\mathcal{T}} = \{(x_i, y_i)|_{i=1}^N\}$: Initial training data, L: number of TICL iterations, $p(\cdot)$: ICL prompt formatting function, S: $\{(\tilde{y}, e)\}$ where $\tilde{y}$: negative sample and e: explanation.

Output: Augmented data $\mathcal{D}^{\text{TICL}}$

1   $\mathcal{D}^{\text{TICL}} \leftarrow \mathcal{D}^{\mathcal{T}}$
2   **for** $l \leftarrow 0$ **to** L **do**
3   **for** $i \leftarrow 0$ **to** $|\mathcal{D}^{\mathcal{T}}| - 1$ **do**
4   // Sample ICL examples excluding x_i (§2.1)
5       $\mathcal{D}^{\text{icl}} = \text{random.sample}(\mathcal{D}^{\text{TICL}} \setminus \{(x_i, y_i, S_i)\})$
6   // Generate model output for x_i (§2.2)
7       $\tilde{y}_i \sim \pi_\theta(p(x_i, \mathcal{D}^{\text{icl}}))$
8   // Generate explanation and determine whether it's a negative sample (§2.3)
9       $(e_i, v_i) \sim \pi_\theta(\mathcal{E}(y_i, \tilde{y}_i))$
10   **if** $v_i \neq$ *True* **then**
11   // Update $\mathcal{D}^{\text{TICL}}$ with the negative sample and explanation
12        $S_i' \leftarrow S_i \cup \{(\tilde{y}_i, e_i)\}$
13        $(x_i, y_i, S_i) \in \mathcal{D}^{\text{TICL}}, (x_i, y_i, S_i) \mapsto (x_i, y_i, S_i')$
14   **return** $\mathcal{D}^{\text{TICL}}$

---

3 Experimental Setup

3.1 Evaluation Dataset

We build on the evaluation setup from DITTO (Shaikh et al., 2024), which uses two public authorship attribution (AA) datasets, because they also study personalized alignment. AA is a classification task of predicting the correct author for some written document, but it can also be reformulated as a personalized alignment task of generating text that is attributed to a specific author. Same as DITTO, we consider 20 authors from two sources: (*i*) CMCC (Goldstein et al., 2008), which contains emails and essays on controversial topics written by students, and (*ii*) CCAT (Lewis et al., 2004), which consists of news articles written by journalists. We preprocess the data with the same steps as DITTO to simulate a realistic data scenario for personalization in which per-user data is limited and treated as private. The number of samples per author and per task is small (Zhang et al., 2020), limited to the smallest number of samples available for an author (12) across both datasets to control for data size. In addition, each author's data is used in isolation such that performance does not depend on the total amount of data available among all authors. The train/val/test split is set to 7/2/3.
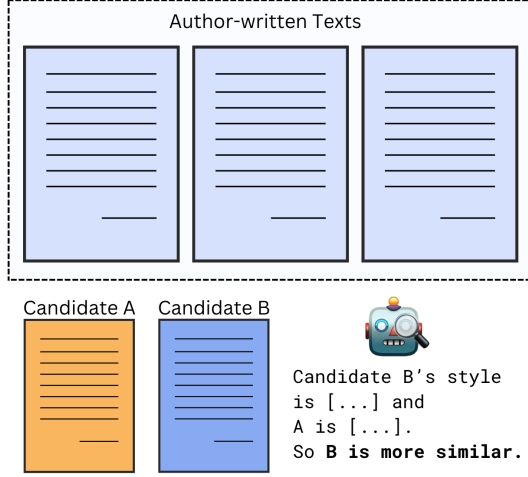
Figure 3: Simplified illustration of our evaluation setup for pairwise comparison of stylistic similarity using LLM-as-a-judge. The full prompt is show in Figure 7.

3.2 Evaluation Method

Pairwise comparison on stylistic similarity with LLM-as-a-judge We evaluate model performance with pairwise comparisons of generated text using LLM-as-a-judge (Zheng et al., 2023; Kim et al., 2023), asking which of the given candidates is more similar to a set of user's texts. A simplified illustration of our evaluation approach is shown in Figure 3.

Specifically, we ask GPT-4o[3] which of two candidates outputs generated for an author's prompt in the test set is more stylistically consistent with five examples sampled from the author's training data. This is similar to the method used in DITTO (Shaikh et al., 2024), but we found that we can further increase accuracy by providing more examples, providing a list of style elements to focus on, and generating an explanation before making its final decision. The two main comparisons we are interested in are how our methods and baselines compare with DITTO's outputs and the author's text. The former is a comparison against the prior state-of-the-art and the latter is against the gold output, and therefore serves as an upper bound. In other words, achieving a $\geq 50\%$ win rate against the author indicates that the method is able to produce outputs that are indistinguishable from the author's texts for our LLM-as-a-judge in terms of stylistic consistency.

Benchmark	Config	Acc.
CMCC	all	$89.5_{2.3}$
	Top 10	$96.6_{1.0}$
CCAT	all	$90.1_{2.6}$
	Top 10	$98.1_{0.7}$
	all, TF-IDF-controlled	$80.5_{2.0}$
	Top 10, TF-IDF-controlled	$96.8_{1.2}$

Table 1: Accuracy for our LLM-as-a-judge setup on author texts. We use the top 10 authors who are most reliably identified by our LLM-as-a-judge to get more reliable evaluation results. Results for individual authors are shown in in appendix in Table 11.

Evaluating LLM-as-a-judge for pairwise comparisons of stylistic similarity[4]. We validate the accuracy of our evaluation method by benchmarking it with user data in CMCC and CCAT. For CMCC, one of the two candidate outputs is the actual text from the correct author and the other is from another human author who wrote a response for the same prompt. This way we can prevent the LLM-as-a-judge from using topical similarity for determining correct authorship, since both candidates will discuss similar content (Wegmann et al., 2022). For CCAT, there are no texts written for the same prompt, so we find the most similar distractor among texts from all other authors using TF-IDF. With this setup, our pairwise comparison evaluation with LLM-as-a-judge achieves $\sim 97\%$ accuracy for both CMCC and CCAT for the top 10 authors, as shown in Table 1.

Author selection Based on these benchmarking results, we discover significantly lower accuracies for some authors, as shown by the large drop in accuracy in Table 1 when all authors are considered. We hypothesize that these authors have less consistent style across their texts, which may cause inconsistent evaluation results. Therefore, instead of random sampling, we select the top 10 authors in terms of our LLM-as-a-judge's accuracy from each dataset for a total of 20 authors.

---

[3]The version we use is `gpt-4o-2024-0806`.

[4]We experiment with various automatic evaluation metrics, human evaluation and alternative LLM-as-a-judge setups, and find pairwise comparisons to be the most reliable. Refer to Appendix B for these results. We intentionally avoid human evaluation based on our own results of human performance on comparing style similarity and evidence from previous work have shown that model-based classifiers are more reliable at determining stylistic similarity than untrained humans (Hallinan et al., 2023; Krishna et al., 2020; Liu et al., 2024; Liu and May, 2024)

	Method	Win rate vs. DITTO		Win rate vs. Author	
		CMCC	CCAT	CMCC	CCAT
Mistral	DITTO	-	-	$25.00_{1.08}$	$10.00_{0.55}$
GPT-4o	Zero-shot	$9.00_{0.72}$	$21.00_{1.05}$	$2.50_{0.34}$	$6.00_{0.40}$
	Few-shot	$42.50_{1.13}$	$76.50_{1.25}$	$21.00_{1.28}$	$25.50_{0.56}$
	CoT	$41.00_{1.01}$	$74.50_{1.04}$	$12.00_{0.53}$	$21.50_{0.47}$
	OPRO	$3.00_{0.56}$	$13.50_{1.27}$	$0.00_{0.00}$	$0.00_{0.00}$
	TICL	$\mathbf{53.00_{1.24}}^\dagger$	$\mathbf{82.50_{1.25}}^\dagger$	$\mathbf{31.00_{1.18}}^\dagger$	$\mathbf{28.00_{0.63}}^\dagger$
Claude 3 Sonnet	Zero-shot	$12.00_{1.03}$	$54.00_{1.44}$	$0.50_{0.11}$	$10.50_{0.66}$
	Few-shot	$74.00_{1.51}$	$86.50_{0.69}$	$43.50_{1.37}$	$53.50_{0.94}$
	CoT	$71.50_{1.20}$	$90.00_{0.67}$	$44.00_{0.88}$	$51.00_{0.81}$
	OPRO	$2.00_{0.25}$	$23.50_{1.33}$	$0.00_{0.00}$	$0.00_{0.00}$
	TICL	$\mathbf{79.50_{1.26}}^\dagger$	$\mathbf{91.50_{0.74}}$	$\mathbf{54.50_{1.67}}^\dagger$	$\mathbf{57.00_{0.87}}^\dagger$

Table 2: Win rates and their corresponding standard errors as subscripts against DITTO (Shaikh et al., 2024) and the original author's response. † indicates a win rate that is larger than the next best performing baseline at a $p < 0.05$ statistically significant level. TICL outperforms all baselines at a statistically significant level, except for Claude 3 Sonnet on CCAT when compared to CoT. Win rate is averaged across 10 authors from each dataset. Full results with per-author scores are in Table 8 and Table 9.

3.3 Baselines

To measure the relative performance of TICL, we also evaluate on (*i*) **Zero-shot**, to capture the base model's vanilla behavior when it is only given the task as input, and (*ii*) **ICL**, where N samples of task and output pairs of the author are included in the context as examples to learn from, along with some guidance to focus on style elements.[5]

In addition, we evaluate other methods that scale test-time compute to improve downstream performance. (*iii*) **Chain-of-Thought** (Wei et al., 2022) is a prompting method that asks to reason about the style elements in the few-shot examples and then generate the response according to the analyzed style and the few-shot examples. (*iv*) **OPRO** (Yang et al., 2024)is a prompt optimization method that refines the prompt iteratively using chain-of-thought prompt as the initial prompt. The model self-updates its prompt based on the history of previous prompts and their scores as "gradients" that guide the update. The initial prompt is set to "Let's think step by step" (Kojima et al., 2022) and the scores are cosine similarity between the Style Embeddings (SE) (Wegmann et al., 2022) of the generated outputs and the reference texts of the validation set. The prompt with the highest score is used at test time.

We also compare TICL to (*v*) **DITTO** (Shaikh et al., 2024), the state-of-the-art that uses TEFT on a small model, i.e. Mistral 7B Instruct (Jiang et al., 2023), and outperforms GPT-4 combined

with few-shot ICL. We do not include SFT in our experiments because DITTO was already shown to be superior to SFT. Lastly, we also compare to (*vi*) **Author**, the author's actual response to the given prompts and thus the upper bound. For (*i*)-(*iv*), we use `gpt-4o-2024-08-06` (Achiam et al., 2023) and `Claude 3 Sonnet` (Anthropic, 2024), abbreviated henceforth as GPT-4o and Claude 3 S unless otherwise specified. We let DITTO and TICL to run for four epochs. For TICL, we use LLM-as-a-judge between outputs from the validation set to choose the best performing prompt. We use Mistral 7B Instruct v0.2 for DITTO, following the original implementation.

4 Results and Discussion

4.1 Main Results

The main evaluation results that compare the baselines against DITTO and against the author's actual reference text is shown in Table 2. Details on the sample size can be found in Appendix A.1.

Vs. DITTO TICL outperforms or performs on par with all other baselines and often outperforms DITTO by a significant margin. We find that explicitly analyzing writing style in the few-shot examples and asking the model to abide by these styles did not lead to consistent improvements. With the exception of Claude 3 Sonnet on CCAT, CoT actually underperforms the simpler few-shot setup. Interestingly, we find that OPRO consistently performs worst. When we analyzed the optimization sequence, OPRO lands on prompts that look sensi-

Model	Ablation	Win rate
GPT-4o	TICL	$31.00_{1.18}$
	− Initial ICL examples	$28.50_{1.50}$
	− Explanations	$23.50_{1.24}$
	− Checkpointing	$22.50_{0.90}$
	− Negative samples & Expl.	$21.00_{1.28}$
Claude 3 S	TICL	$54.50_{1.67}$
	− Initial ICL examples	$52.00_{1.22}$
	− Explanations	$46.00_{1.41}$
	− Checkpointing	$54.00_{1.67}$
	− Negative samples & Expl.	$43.50_{1.37}$

Table 3: Ablation with TICL minus (−) individual procedures on CMCC. Reported win rate is vs. Author and subscript is the standard error. All procedures provide performance gains, but explanations make the biggest difference while starting with ICL examples is the least important.

ble, e.g. *"Respond with rhetorical devices, simple sentence structures, and easy vocabulary..."*, but these generalized guidance are not as effective as seeing few-shot examples within the context in producing stylized text.

Vs. Author   We see similar strengths for TICL when compared to Author, for which the results are shown in Table 9. Surprisingly, Claude 3 Sonnet is able to exceed the ≥ 50 win rate threshold when using TICL for both CMCC and CCAT. Performance with GPT-4o is relatively lower, but still significantly higher than DITTO for both datasets. We further dive into why Claude 3 Sonnet outperforms GPT-4o in §4.3

Given Claude's strong performance even in the zero-shot setting against DITTO, we suspect there may be some leakage of CCAT data into Claude. Therefore, we focus our analysis henceforth mainly on CMCC. Also, since the relative performance among models is similar between vs. DITTO and vs. Author results, we focus on vs. Author results in the following sections.

4.2   Ablating TICL

We conduct an ablation study to understand how each of the components in TICL contributes to its performance. We observe how the win rates change when we remove the initial ICL setup that substitutes SFT for behavior cloning, explanations, checkpointing (choosing the best performing augmented prompt throughout iterations rather than simply using the final prompt), and both the negative samples and explanations. The last row is equivalent to the few-shot baseline. The results are

Model	Swap Configuration	Win rate
GPT-4o	No swap	$31.00_{1.18}$
	→ with Claude 3 S expl.	$32.50_{1.13}$
	→ with Claude 3 S TICL	$18.30_{0.25}$
Claude 3 S	No swap	$54.50_{1.67}$
	→ with GPT-4o expl.	$55.00_{1.28}$
	→ with GPT-4o TICL	$42.50_{0.92}$

Table 4: Ablation with various swap configurations to compare GPT-4o TICL and Claude 3 Sonnet TICL performance on CMCC. Result format is the same as Table 3. Changing the explanation model does not lead to significant changes. Using final TICL prompts from a different model leads to large drops, indicating that models learn better from their own failure modes.

shown in Table 3.

Explanations are most important, while starting with ICL provides smallest improvements. For both GPT-4o and Claude 3, removing any of the procedures leads to performance drops. For both cases, we see the largest drop attributed to removing both the explanations and negative samples. When only dropping the explanations, we see a smaller drop, indicating that both the negative samples and explanations provide additive advantages. For GPT-4o, checkpointing is also very important, but less so for Claude 3 Sonnet. In both models, not having the first step use ICL examples is not critical, meaning that we can even start with zero-shot outputs for the first step and not see large performance degradation.

4.3   Why does Claude 3 Sonnet outperform GPT-4o?

One striking trend from Table 2 is Claude 3 Sonnet's noticeably stronger performance compared to GPT-4o. This is somewhat surprising given that GPT-4o is the latest model that ranks at the top of popular language modeling leaderboards, such as Chatbot Arena.[6] In order to understand Claude's comparative performance over GPT-4o, we examine how performance changes if we (*i*) use the other model for generating explanations during prompt augmentation (with other model expl.) and (*ii*) use the fully developed TICL prompt from the other model at test time (with other model TICL).

Explanations are similar in quality.   Results shown in Table 4 suggest that it is **not** that Claude 3 Sonnet generates better explanations than GPT-4o.

---

[6] https://lmarena.ai/

A	B	FRE (A \ B) ↑	FRE (B \ A) ↑	n-gram	z-score	p
GPT-4o TICL	GPT-4o Few-shot	121.22	36.62	so why	1.990	0.047
				honestly	2.098	0.036
				additionally	−2.983	0.003
				therefore	−2.233	0.026
Claude 3 S TICL	GPT-4o TICL	120.21	77.21	believe	4.255	0.000
				feel that	4.195	0.000
				crucial to	−3.449	0.001
				implications of	−2.925	0.003

Table 5: Representative Fightin' Words model (Monroe et al., 2008) results for frequency differences between two sources at $p < 0.05$. A positive z-score indicates that output source A significantly uses more of the n-gram than B while a negative score signals the opposite. We observe that the models that have higher win-rates from pairwise comparisons frequently generate phrases for expressing personal opinions while the losing model generates more formal phrases.

Applying TICL on GPT-4o with Claude 3 Sonnet as the explanation generator leads to a mild increase in performance that is not statistically significant. This is further corroborated by Claude 3 Sonnet's results when using GPT-4o as the explanation generator, which also leads to a statistically insignificant improvement.

TICL prompts are not transferable. Using the final augmented prompts developed by Claude 3 Sonnet and using GPT-4o at test-time does not improve on simply using GPT-4o's own augmented prompt. This is likely due to the distribution shift: negative samples and their corresponding feedback is not as relevant and actually serves as noise such that the performance actually drops below the few-shot performance in Table 2. Based on these results, we attribute Claude 3 Sonnet's superior performance on its superior few-shot ICL capabilities for writing tasks. The following section on lexical analysis adds support to this hypothesis.

4.4 Lexical Analysis

We further explore the models' comparative performances using a lexical analysis, using Fightin' Words model for extracting words with significant differences in between two corpora (Monroe et al., 2008). The method shows different ratios of words that are significantly more frequent in one model over another. Using a p-level of 0.05, we calculate the number of n-grams that are significantly more frequent in the author's text compared to the model's text and vice-versa. Representative results are shown in Table 5.

TICL overcomes model bias for structural and formal phrases better than ICL. Comparing GPT-4o TICL vs. GPT-4o few-shot elicits TICL

improvement over a simple ICL baseline. We see that structural and formal phrases, e.g. "*additionally*" and "*therefore*", are more frequent in the few-shot outputs, while TICL contain more colloquial phrases, e.g., "*so why*". This suggests that TICL enables models to overcome its biases towards generic formal text and better learn from the given style context.

Claude 3 Sonnet applies user styles more effectively than GPT-4o. When comparing Claude 3 Sonnet and GPT-4o directly, the top words that Claude 3 Sonnet is able to generate more consistently are subjective and assertive words or phrases that are frequently seen in CMCC, e.g., "*believe*" and "*feel that*", and also more casual phrases, e.g., "*things like*" and "*kind of.*" On the other hand, GPT-4o's outputs consists of more formal phrases, such as "*crucial to.*" This dynamic is further corroborated by the Flesch Readability Ease (FRE) scores (Flesch, 1948). The words that appear significantly more frequently for Claude has a higher FRE score (FRE(A\B)=120.21) than the score for words that appear significantly more frequenty for GPT-4o (FRE(B\A=77.21). A higher FRE score means it is easier to understand when read, which correlates to more casual text (Cho et al., 2024).

5 Related Work

5.1 Personalization

Personalization is an overloaded term in the context of language models as it can refer to multiple distinct features. Language models can be personalized to complete tasks as a specific user (Salemi et al., 2023; Zhuang et al., 2024), answer questions according to a user's profile (Kim and Yang, 2024; Dutt et al., 2022), and recommend content based

on a user's knowledge level (Wang et al., 2016). They can also respond in a way that aligns with an individual's preference of digital assistance behavior (Jang et al., 2023), edit an existing text to match someone else's style (Horvitz et al., 2024; Patel et al., 2023) . and generate some output in the style of a target user, such as a response to another online user (Liu et al., 2023) or text for a writing task (Shaikh et al., 2024; Kumar et al., 2024; Li et al., 2024; Mysore et al., 2023).

TICL focuses on the last category mentioned above – personalized text generation – where we want to steer language models away from a generic output (Li et al., 2016; Zhang et al., 2021; Padmakumar and He, 2024) for a given writing task and instead produce one that is specific to a user's style (Rivera-Soto et al., 2021; Wegmann et al., 2022). Prior work on personalized text generation rely on at least one component that requires fine-tuning, either the model that generates the response (Li et al., 2023; Liu et al., 2023; Shaikh et al., 2024; Mysore et al., 2023) or a model that revises the prompt passed on to a black box model (Li et al., 2024). However, TICL is an inference-only method that does not require any parameter updates or new parameters.

5.2 Inference-only learning

One major branch of inference-only learning method focuses on automating prompt engineering, optimizing the wording of a prompt template that is placed prior to asking a model to complete a task (Shin et al., 2020; Zhou et al., 2023; Ma et al., 2024; Yang et al., 2024; Kim and Yang, 2024; Ye et al., 2024), e.g. *"Let's think step by step"* from (Kojima et al., 2022). Recent work have operationalized this process to also automate example selection for few-shot examples (Yuksekgonul et al., 2024; Khattab et al., 2023). In contrast, TICL maximizes its understanding of the target task by iteratively augmenting the content used for ICL with contrastive examples and explanations while keeping the prompt template constant.

Another prominent branch of inference-only learning is agentic methods that build on intermediate LLM outputs to refine the final output (Chen et al., 2023; Wei et al., 2022; Yao et al., 2023; Sumers et al., 2024; Shinn et al., 2024; Saha et al., 2024). Reflexion (Shinn et al., 2024) is most similar to our work, but it cannot be directly applied without a reliable evaluator for measuring success.

Another key difference with TICL is that it front-loads the trial-and-error process ahead of test time and does not augment the prompt a priori by reasoning over few-shot examples.

6 Conclusion

In conclusion, we develop TICL, a tuning-free method for personalized alignment that adapts prior work on trial-and-error fine-tuning with scaling inference compute. Instead of fine-tuning a model with synthetic negative samples, TICL uses them to augment the prompt and generate explanations that provide further fine-grained guidance on how to align outputs towards a desirable style. On personalized text generation tasks, TICL outperforms other competitive tuning-free baselines and the previous state-of-the-art. TICL provides an approach for leveraging black box models for personalization without any fine-tuning.

Limitations

One of the limitations of TICL is that it is more computationally costly than fine-tuned models at test time because of the longer inputs due to the few-shot examples and their corresponding negative samples and explanations. However, inference cost for repeating prompts can be significantly lowered by prompt caching.[7] In addition, inference costs in general are continuously getting lower with optimization efforts that comprehensively span hardware[8] and software (Kwon et al., 2023). As the promise of inference-only methods become increasingly evident, we believe these efforts will be further accelerated and lead to even more favorable conditions for methods such as TICL.

Another limitation of TICL is that it is only effective for models that are able to understand long contexts well. In our preliminary experiments, our results on using TICL with smaller models such as Mistral 7B Instruct and GPT-4o mini were poor as these models continued to generate outputs similar in style to what were labeled in-context as bad examples, indicating that they were not able to distinguish between the good and bad examples included in the context. Therefore, the performance of TICL is dependent on a model's long context understanding.

---

[7]https://www.anthropic.com/news/prompt-caching

[8]Groq and AWS Inferentia are some examples of inference-specific hardware development.

Lastly, TICL results in per-user or per-task prompts, which is similar to the limitations of prior work that rely on fine-tuned per-user model parameters. Whether we can use TICL to enable a single model to perform personalized task completion for multiple tasks without requiring excessively long prompts by (*i*) discovering methods that maintain or improve performance while compressing the prompt or (*ii*) injecting the information in the prompt for controlling style during decoding is an exciting avenue for future work.

Generative AI Statement

No generative AI tools were used in the writing of this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. Self-ICL: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662, Singapore. Association for Computational Linguistics.

Hyundong Cho, Nicolaas Jedema, Leonardo F. R. Ribeiro, Karishma Sharma, Pedro Szekely, Alessandro Moschitti, Ruben Janssen, and Jonathan May. 2024. Speechworthy instruction-tuned language models.

Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiah, Dan Roth, and Carolyn Rose. 2022. PerKGQA: Question answering over personalized knowledge graphs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 253–268, Seattle, United States. Association for Computational Linguistics.

Todd Firsich and Anthony Rios. 2024. Can GPT4 detect euphemisms across multiple languages? In *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, pages 65–72, Mexico City, Mexico (Hybrid). Association for Computational Linguistics.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Jade Goldstein, Kerri Goodwin, Roberta Sabin, and Ransom Winder. 2008. Creating and using a correlated corpus to glean communicative commonalities. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Skyler Hallinan, Faeze Brahman, Ximing Lu, Jaehun Jung, Sean Welleck, and Yejin Choi. 2023. STEER: Unified style transfer with expert reinforcement. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7546–7562, Singapore. Association for Computational Linguistics.

Zachary Horvitz, Ajay Patel, Kanishk Singh, Chris Callison-Burch, Kathleen McKeown, and Zhou Yu. 2024. TinyStyler: Efficient few-shot text style transfer with authorship embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13376–13390, Miami, Florida, USA. Association for Computational Linguistics.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Jaehyung Kim and Yiming Yang. 2024. Few-shot personalization of llms with mis-aligned responses.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chein Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. Longlamp: A benchmark for personalized long-form text generation.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2024. Learning to rewrite prompts for personalized text generation. In *Proceedings of the ACM on Web Conference 2024*, pages 3367–3378.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach llms to personalize–an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 110–119, San Diego, California. Association for Computational Linguistics.

Shuai Liu, Shantanu Agarwal, and Jonathan May. 2024. Authorship style transfer with policy optimization.

Shuai Liu, Hyundong Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. 2023. RECAP: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8404–8419, Toronto, Canada. Association for Computational Linguistics.

Shuai Liu and Jonathan May. 2024. Style transfer with multi-iteration preference optimization.

Ruotian Ma, Xiaolei Wang, Xin Zhou, Jian Li, Nan Du, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Are large language models good prompt optimizers?

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2023. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. *arXiv preprint arXiv:2311.09180*.

Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? In *The Twelfth International Conference on Learning Representations*.

Ajay Patel, Nicholas Andrews, and Chris Callison-Burch. 2023. Low-resource authorship style transfer: Can non-famous authors be imitated?

Richard Plant, Valerio Giuffrida, and Dimitra Gkatzia. 2022. You are what you write: Preserving privacy in the era of large language models. *arXiv preprint arXiv:2204.09391*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus

Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024. Branch-solve-merge improves large language model evaluation and generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8352–8370, Mexico City, Mexico. Association for Computational Linguistics.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. Lamp: When large language models meet personalization. *arXiv preprint arXiv:2304.11406*.

Omar Shaikh, Michelle Lam, Joey Hejna, Yijia Shao, Michael Bernstein, and Diyi Yang. 2024. Show, don't tell: Aligning language models with demonstrated feedback. *arXiv preprint arXiv:2406.00888*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7584–7600, Bangkok, Thailand. Association for Computational Linguistics.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2024. Cognitive architectures for language agents. *Transactions on Machine Learning Research*. Survey Certification.

Franck Tétard and Mikael Collan. 2009. Lazy user theory: A dynamic model to understand user selection of products and services. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–9. IEEE.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lin Wang, Bin Wu, Juan Yang, and Shuang Peng. 2016. Personalized recommendation for new questions in community question answering. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 901–908. IEEE.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. 2024. Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 355–385, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

Zhipeng Zhang, Yasuo Kudo, Tetsuya Murai, and Yong-gong Ren. 2020. Improved covering-based collaborative filtering for new users' personalized recommendations. *Knowledge and Information Systems*, 62:3133–3154.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li13, Eric P Xing35, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Pei Zhou, Hyundong Cho, Pegah Jandaghi, Dong-Ho Lee, Bill Yuchen Lin, Jay Pujara, and Xiang Ren. 2022. Reflect, not reflex: Inference-based common ground improves dialogue response quality. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10450–10468, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. 2024. Hydra: Model factorization framework for black-box llm personalization.

Appendix

A   Evaluation Dataset Details

A.1   Sampling

We rely on sampling to alleviate issues of having a small test set per author. For each of the three test prompts, we sample five outputs per method. We pair outputs for the same prompts to create 75 comparison pairs ($5 \cdot 5 \cdot 3$) and sample 40 of them for comparison and randomly assign the order they appear in the evaluation prompt to control for order bias. For the versus author results, we only have 15 samples since there's only one original author response ($1 \cdot 5 \cdot 3$), so we permute the order for each pair to get 30 samples. This results in 400 ($40 \cdot 10$) comparisons for vs DITTO and 300 ($30 \cdot 10$) for vs Author to get the averaged win rates in Table 2. We are able to compute statistically significant win rates with these number of comparisons, as reported in Section 4.1.

A.2   Technical Details

The exact models that we use for our experiments are `gpt-4o-2024-0806`, Claude 3 Sonnet and Mistral 7B Instruct v0.2 via Amazon Bedrock. For evaluation (§3.2), we use batch inference using the

Benchmark	Author IDs									
CMCC	2	3	5	6	7	8	11	13	15	17
CCAT	10	12	15	20	23	27	28	30	32	38

Table 6: Mapping from author indices to actual author IDs in CMCC and CCAT that we use throughout our work. These authors are the authors that ranks in the top 10 for highest accuracy achieved with GPT-4 Eval, as described in Appendix B.2 and shown in Table 11.

OpenAI API, which lets us reduce inference costs by 50%.[9]

Notes on TICL   At the first step of TICL, we hold out one sample (x_i, y_i) at random and use the rest $\mathcal{D}_0^{\mathcal{T}} \setminus \{(x_i, y_i)\}$ as few-shot examples that form the ICL prompt. If we use the full set, we don't have any samples for which we can explore and learn without seeing the reference output y_i of the input x_i. from $\mathcal{D}_0^{\mathcal{T}}$. This information is made more explicit in Algorithm 1.

Notes on DITTO.   DITTO was trained with Mistral 7B Instruct, which is suspected to be a smaller model than Claude 3 Sonnet and GPT-4o. We found that the fine-tuned variants of Mistral 7B Instruct are prone to generating unstable outputs, producing template components such as `[INST]` or generating degenerate outputs that contain repetitive content. To give these approaches the best chances possible against our approaches with larger models, we reject samples that contain template components and repetitive content until we reach the desired number of samples $N = 5$ from each model.

B   Evaluation Methods

In this section, we describe benchmarking results using author texts from CMCC and CCAT for various evaluation methods. Results are shown in Table 7. For CMCC, several authors wrote responses to the same prompt, and therefore we use samples written for the same prompt as the compared samples, as this would help us control for content similarity between the tested sample and the examples in making the task arbitrarily easy.

For CCAT, which are news articles with a format of "*Write a news article that starts with the following sentence*: `article's first sentence`". There are no share prompts for this dataset. Since journalists tend to write articles on

---

[9]`https://platform.openai.com/docs/guides/batch`

Method	# Examples	Explan.	CMCC	CCAT
GPT-4o pairwise	1	✗	$76.1_{2.2}$	-
	1	✓	$78.1_{2.1}$	-
	3	✓	$82.6_{1.9}$	-
	5	✓	$89.5_{2.3}$	$80.4_{2.0}$
GPT-4o mini pairwise	5	✓	$68.2_{2.4}$	$66.8_{2.3}$
GPT-4o rating	5	✓	$55.0_{1.2}$	$20.0_{1.1}$
SE	1	-	$71.5_{1.5}$	$61.3_{1.5}$
	3	-	$76.8_{1.4}$	$63.0_{1.5}$
	5	-	$78.5_{1.3}$	$63.6_{1.5}$
Human	5	-	85.4	-

Table 7: Results with author texts from CMCC and CCAT for comparing various evaluation methods. GPT-4o: `gpt-4o-2024-0806`, GPT-4o mini: `gpt-4o-2024-0718-mini`, SE: Style Embeddings.

similar topics over time and thus content similarity can be an easy hint for detecting authorship, we select the negative sample as one that has the highest TF-IDF similarity with any one of the in-context examples.

B.1 Embedding-based evaluation

Universal authorship representations (UAR) (Rivera-Soto et al., 2021) is a sentence embedding method that uses a SBERT model (Reimers and Gurevych, 2019) fine-tuned with a contrastive learning objective such that the representations of sentences or documents from the same author become closer together than with those of other authors. Style Embeddings (SE) is a follow-up to UAR that refines the representations to focus on style rather than content by pairing contrastive samples that share similar content (Wegmann et al., 2022), and therefore we focus on embedding-based comparisons with Style Embeddings.

We compute the cosine similarity between candidate texts' SE with those of the author examples and the one with higher cosine similarity is considered more stylistically similar. The best score for SE is 78.5% and 63.6% for CMCC and CCAT, respectively, which is significantly lower than GPT-4o results but better than GPT-4o mini's results. While having more examples also helps with SE's performance, it plateaus markedly at three examples, and there is only minimal gains observed for CCAT.

B.2 LLM-as-a-Judge

The pairwise comparison setup is described in Section 3.2. As an alternative, we also consider having LLM-as-a-judge provide ratings for individual instances, so that we can reduce the number of samples that the LLM-as-a-judge need to evaluate and also measure relative performance of each apporach based on their aggregate scores. The prompt is similar to the pairwise comparison setup except that only one candidate is shown and the LLM is asked to provide a score from 1-5 on how stylistically similar it thinks the candidate is to the examples. Unfortunately, this result is the poorest performing approach, only achieving 55.0% accuracy for CMCC and 20.0% accuracy for CCAT. The main reason for low accuracy is that the majority of instances were given the same rating such that they were deemed as ties.

B.3 Human Evaluation

We have three human annotators complete the same setup with 100 samples for CMCC only with five samples from each author. Conducting the same evaluation for CCAT was overwhelming for human participants because the average text length in CCAT is much larger. As shown in the results, human performance is lower than the best GPT-4o results. This reinforces findings from previous work that identifying authorship reliably is a difficult task for untrained humans and that model-based classifiers are more reliable and practical for this task (Hallinan et al., 2023; Krishna et al., 2020; Liu et al., 2024; Liu and May, 2024).

The full prompt template for this evaluation setup is shown in Figure 7.

C Flesch Reading Ease

The Flesch Reading Ease (FRE) score is a simple equation for approximating readability of a given text based on the average number of words per sentence and the average number of syllables per word (Flesch, 1948). The formula for calculating FRE is the following:

$$FRE = 206.835 - (1.015 \cdot ASL) - (84.6 \cdot \frac{n_{sy}}{n_w})$$

where ASL is average sentence length ($\frac{\text{total words}}{\text{total sentences}}$) and $\frac{n_{sy}}{n_w}$ is the average number of syllables per word ($\frac{\text{total syllables}}{\text{total words}}$). The higher the score, the easier it is to understand a piece of text

when read. The maximum score is 121.22, while there is no limit to how low it can be and it can even be negative. We use Python's `textstat` package[10] to compute FRE.

D Full Per-Author Results

The full per-author results with comparisons against DITTO and against the author's text are shown in Table 8 and Table 9, respectively.

E Sample Outputs

We show some qualitative examples from our baselines and TICL in Table 10.

F Prompt Details

For details on the prompts that we use, we refer the reader to the following:

- Few-shot prompting: Figure 5
- TICL: Figure 6
- Pairwise LMJ evaluation: Figure 7
- Style guide generation for CoT: Figure 8
- Writing prompt for CoT that uses the generated style guide: Figure 9
- OPRO's optimization prompt: Figure 10
- OPRO's writing prompt that uses the prompt found from the optimization process: Figure 11

---

[10] https://github.com/textstat/textstat

```
You are an editor.

    - Your task is to analyze whether the candidate writing is stylistically consistent
      with the author's writing(s) and if not, highlight elements of the author's style
      that are not observed in the candidate writing.
    - Consider similarity with regards to the (1) length, (2) format, (3) paragraph
      structure, (4) sentence structure, (5) punctuation, (6) syntax, (7) voice, and (8)
      diction of the author's writing, but NOT the content it covers.
    - Use the minimum words possible in your analysis while providing specific examples
      of how the observed inconsistencies must be edited to become stylistically
      consistent with the author's writing.
    - If the candidate writing is stylistically consistent with the author's writing,
      respond with "yes" in the "is_consistent" field.  Otherwise, respond with "no".


# Task
{ task }


# Author's writing
{ reference_text }


# Candidate writing to edit
{ generated_text }


Respond only with JSON with the following format:
{
  "explanation":  "<style analysis and suggested edits>",
  "is_consistent":  "<yes/no>"
}
```

Figure 4: Prompt template $\mathcal{E}$ for generating explanations on the difference between the reference output and the generated output.

```
You are a stylistically consistent writer.  Below are examples that exemplify your
writing style.

# Writing Task Example 1
{ example["task"] }
## Your Writing 1
{ example["reference_output"] }

# Writing Task Example 2
...


** Task to complete **
Now complete the following writing task with a style and format consistent with 'Your
Writing' examples.
Be consistent in terms of (1) length, (2) format, (3) paragraph structure, (4) sentence
structure, (5) punctuation, (6) syntax, (7) voice, and (8) diction of your writing when
completing the task.


Task:  { target_task }

Directly provide your response in the following format:


```
<your writing>
```
```

Figure 5: Prompt template $\mathcal{P}$ for the few-shot in-context learning baseline.

```
You are a stylistically consistent writer.  Below are examples that exemplify your
writing style.

# Writing Task Example 1
{ example["task"] }
## Your Writing 1
{ example["reference_output"] }

## Stylistically Inconsistent Writing 1-1
{ example["generated_output"][0]["output"] }
### Inconsistent stylistic elements in `Stylistically Inconsistent Writing 1-1` that
should be corrected for it to become more consistent with `Your Writing 1`:
{ example["generated_output"][0]["explanation"] }
...

# Writing Task Example 2
...


** Task to complete **
Now complete the following writing task with a style and format consistent with
`Your Writing` examples and also avoiding the stylistic inconsistencies found in the
`Stylistically Inconsistent Writing` examples.
Be consistent in terms of (1) length, (2) format, (3) paragraph structure, (4) sentence
structure, (5) punctuation, (6) syntax, (7) voice, and (8) diction of your writing when
completing the task.


Task:  { target_task }

Directly provide your response in the following format:
```
<your writing>
```
```

Figure 6: Prompt template $\mathcal{P}$ for TICL. It builds on Figure 5 by adding negative samples and explanations generated with Figure 4.

```
You are an impartial evaluator of style similarity.  Below are samples of an author's
writing and two options.

# Author's Writing:
EXAMPLE 1:
{example 1}
...
EXAMPLE 5:
{example 5}

# Option A:
{candidate A}

# Option B:
{candidate B}

# Task
Which option is more likely to have been written by the author based on style similarity
to the samples given as AUTHOR'S WRITING above?  Consider each option's similarity with
regards to the (1) length, (2) format, (3) paragraph structure, (4) sentence structure,
(5) punctuation, (6) syntax, (7) voice, and (8) diction of the author's writing, but
NOT the content it covers.  If one option has incoherent/odd text or formatting (e.g.,
random dashes, repetitive text, random signatures, etc.)  that is not present in the
author's writing while the other doesn't, it should be considered less similar.
{
  "explanation":  {
    "length":  "<Explanation>",
    "format":  "<Explanation>",
    "paragraph structure":  "<Explanation>",
    "sentence structure":  "<Explanation>",
    "punctuation":  "<Explanation>",
    "syntax":  "<Explanation>",
    "voice":  "<Explanation>",
    "diction":  "<Explanation>",
    "odd incoherent text/formatting":  "<Explanation>"
  }
  "answer":  "<The option more similar to the AUTHOR'S WRITING; either A or B>"
}
ALWAYS REMAIN IMPARTIAL WHEN EVALUATING OUTPUTS AND PENALIZE ODD INCOHERENT TEXT.
```

Figure 7: Prompt template for evaluating which output is more stylistically similar to the target author's writing style given five examples from the target author.

```
Given the following written examples, provide a style guide on how you would go about
writing in a similar style as the examples for the target task.  Analyze the (1) length,
(2) format, (3) paragraph structure, (4) sentence structure, (5) punctuation, (6)
syntax, (7) voice, and (8) diction of the examples, but NOT the content or topic that
they cover.

# Writing Task Example 1
{ example["task"] }
## Your Writing 1
{ example["reference_output"] }

# Writing Task Example 2
...

# Target task
Task:  { target_task }

Provide your style guide with the following format:
```
<your writing>
```
```

Figure 8: Prompt template for generating style guides that will be used for Chain-of-Thought guided writing in Figure 9.

```
You are a stylistically consistent writer.  Below are examples that exemplify your
writing style.

# Writing Task Example 1
{ example["task"] }
## Your Writing 1
{ example["reference_output"] }

# Writing Task Example 2
...

# Task to complete
Now complete the following writing task, first by analyzing the style and format of the
'Your Writing' examples.

Task:  { target_task }

{ Chain-of-Thought Style Guide }

Directly provide your response for the task in the following format:
```
<your writing>
```
```

Figure 9: Prompt template for Chain-of-Thought-guided writing. The blue text is the style guide generated from Figure 8.

5896

```
I have some instructions along with their corresponding scores.  The instructions are
arranged in ascending order based on their scores, where higher scores indicate better
quality.


{% for prompt_score_pair in prompt_score_pairing %}
instruction:
{{ prompt_score_pair[0] }}
score:
{{ prompt_score_pair[1] }}
{% endfor %}

The following exemplars show how to apply your instruction:  you replace <INS> in each
input with your instruction, then read the input and give an output.  The scores are
the average stylistic similarity score of the generated output, generated with your
instruction, compared to the reference output.

{% for example in examples %}
input:
{{ example["task"] }}
<INS>
output:
{ example["reference_output"] }
{% endfor %}

Write your new instruction that is different from the old ones that will help achieve a
style similarity score as high as possible.  Consider similarity with regards to the (1)
length, (2) format, (3) paragraph structure, (4) sentence structure, (5) punctuation,
(6) syntax, (7) voice, and (8) diction of exemplar outputs, but NOT the content it
covers.  The instruction should not mention 'reference output' or 'original input' as
only the input and the instruction placed instead of <INS> is available when writing an
output.  Write your instruction in the following format:
```
<your instruction>
```
```

Figure 10: OPRO optimization prompt. The instruction with the highest score is used for the OPRO writing prompt in Figure 11.

```
Task:  { target_task }
{ opro_prompt }

Respond only with JSON with the following format:
{
    "thought": "<your thoughts>",
    "response": "<your response>"
}
```

Figure 11: OPRO writing prompt. The blue text is the instruction that is selected from the prompt optimization process from Figure 10.

Data	Method		a_{avg}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
CMCC	GPT-4o	Zero-shot	$9.00_{0.72}$	20	5	0	0	15	10	10	0	30	0
		Few-shot	$42.50_{1.13}$	55	35	40	20	65	55	40	15	55	45
		CoT	$41.00_{1.01}$	50	35	35	25	55	50	50	15	60	35
		OPRO	$3.00_{0.56}$	5	0	0	0	0	0	0	0	25	0
		TICL	$\mathbf{53.00_{1.24}}^{\dagger}$	70	70	50	35	60	65	80	45	70	50
	Claude 3 Sonnet	Zero-shot	$12.00_{1.03}$	25	5	5	0	20	15	5	0	45	0
		Few-shot	$74.00_{1.51}$	85	55	70	35	85	90	95	50	100	75
		CoT	$71.50_{1.20}$	80	60	65	55	70	85	90	45	100	65
		OPRO	$2.00_{0.25}$	5	0	0	0	0	0	5	0	10	0
		TICL	$\mathbf{79.50_{1.26}}^{\dagger}$	85	50	65	90	75	95	90	95	100	100
CCAT	GPT-4o	Zero-shot	$21.00_{1.05}$	15	20	15	5	50	5	30	20	40	10
		Few-shot	$76.50_{1.25}$	90	70	90	40	90	70	95	90	60	70
		CoT	$74.50_{1.04}$	75	65	85	45	85	90	90	90	80	40
		OPRO	$13.50_{1.27}$	0	5	0	0	35	10	55	15	10	5
		TICL	$\mathbf{82.50_{1.25}}^{\dagger}$	85	65	90	70	85	90	100	90	80	70
	Claude 3 Sonnet	Zero-shot	$54.00_{1.44}$	50	40	65	35	65	35	90	80	50	30
		Few-shot	$86.50_{0.69}$	95	80	95	65	95	85	95	85	90	80
		CoT	$90.00_{0.67}$	100	95	100	70	85	85	100	90	90	85
		OPRO	$23.50_{1.33}$	20	5	10	10	55	5	55	15	30	30
		TICL	$\mathbf{91.50_{0.74}}$	100	90	100	75	100	90	100	95	80	85

Table 8: LMJ-based win rate results against DITTO (Shaikh et al., 2024). † indicates a win rate that is larger than the next best performing baseline at a $p < 0.05$ statistically significant level. TICL outperforms all other baselines at a statistically significant level, except for Claude 3 Sonnet on CCAT. a_{avg} is the average win rate across all authors and $a1$-$a10$ are the author IDs from Table 6.

Data	Method		a_{avg}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
CMCC	Mistral	DITTO	$25.00_{1.08}$	20	20	35	45	10	35	5	30	5	45
	GPT-4o	Zero-shot	$2.50_{0.34}$	0	0	15	5	0	0	0	0	5	0
		Few-shot	$21.00_{1.28}$	20	15	10	25	35	15	10	0	15	65
		CoT	$12.00_{0.53}$	25	10	10	10	20	5	10	0	10	20
		OPRO	$0.00_{0.00}$	0	0	0	0	0	0	0	0	0	0
		TICL	$\mathbf{31.00_{1.18}}^{\dagger}$	30	20	15	35	35	25	45	5	35	65
	Claude 3 Sonnet	Zero-shot	$0.50_{0.11}$	0	0	0	0	0	0	0	0	5	0
		Few-shot	$43.50_{1.37}$	60	40	30	25	45	45	55	5	60	70
		CoT	$44.00_{0.88}$	50	40	30	45	25	40	65	35	55	55
		OPRO	$0.00_{0.00}$	0	0	0	0	0	0	0	0	0	0
		TICL	$\mathbf{54.50_{1.67}}^{\dagger}$	80	15	30	75	60	75	65	55	60	85
CCAT	Mistral	DITTO	$10.00_{0.55}$	0	15	15	15	5	25	0	5	10	10
	GPT-4o	Zero-shot	$6.00_{0.40}$	0	10	0	15	10	0	0	10	10	5
		Few-shot	$25.50_{0.56}$	15	15	30	25	35	35	15	30	25	30
		CoT	$21.50_{0.47}$	15	10	15	20	25	30	25	30	20	25
		OPRO	$0.00_{0.00}$	0	0	0	0	0	0	0	0	0	0
		TICL	$\mathbf{28.00_{0.63}}^{\dagger}$	10	30	25	40	45	40	20	30	30	30
	Claude 3 Sonnet	Zero-shot	$10.50_{0.66}$	0	5	10	25	20	0	0	20	15	10
		Few-shot	$53.50_{0.94}$	60	70	40	55	45	40	35	55	60	75
		CoT	$51.00_{0.81}$	55	70	40	50	50	35	45	50	45	70
		OPRO	$0.00_{0.00}$	0	0	0	0	0	0	0	0	0	0
		TICL	$\mathbf{57.00_{0.87}}^{\dagger}$	60	55	45	55	80	40	45	55	70	65

Table 9: LLM-as-a-judge win rates against the author's actual text. The table takes the same format as Table 8.

CMCC	CCAT

Dataset

Write an approximately 500 word essay to the following prompt: Recently, school officials prevented a school shooting because one of the shooters posted a myspace bulletin. Do you think this was an invasion of privacy?	Write an article that starts with the following: A potential new source of revenue – oil – offers hope of a fresh start for Chad's fragile economy as it rebuilds after three decades of war.

Author

[...] I feel as though the school officials were not invading privacy at all. The entire point [...] Why anyone [...] is beyond me, and even more so if the person does not expect to be caught. If anyone [...] Did the school harm the person who was going to do the shooting? No. [...]since it IS available to the public- to TURN OVER [...] If there are lives in danger [...] (to catch kids drinking, etc.), then THAT would be "taking it over the edge" as an invasion of privacy. Thanks for your time, ID # 20	[...] "Soon work will begin," Foreign Minister Saleh Kebzabo told Reuters recently. "Chad should get some royalties so we can free the country through development." Chad has known coups, civil war or conflict with its northern neighbour Libya for much of its life from independence in 1960. Sporadic cross-border rebel raids remain a problem. [...] But [...] On the negative side, external debt rose to $817 million in 1996 from $803 million in 1995 and 1996 GDP growth was projected at 0.5 percent, down from 4.5 percent in 1995. [...]

DITTO

[...] I definitely do not think that the school system invaded the privacy of the person who was caught posting the myspace bulletin about the potential school shooting. [...] If this was the case, then he definitely knows what a good portion of the world is aware of- the fact that everything you do can be caught on camera. Whether it be a store, etc., everything is being recorded. [...] Sincely, ID# 20123452	[...] The landlocked African nation tapped its first well in 1973 but suffered instability through French suitor Cameroon's push for a larger share of any wealth or former guerrilla revolts against president Idriss Deby's rule. "Oil is a godsend. It is not only a source of revenue but it also gives Chad prestige on the international scene," said Jean-Pierre Alain, head of Chad's national hydrocarbons agency. [...] It had an estimated 1997/98 budget of $275 million. It had international debt of $355 million at the end of 1996 [...]

Few-shot

[...] In regards to the question of whether the school officials' actions to prevent a school shooting based on a myspace bulletin were an invasion of privacy, I hold the firm belief that they were not. The fundamental duty of any educational institution [...] Ultimately, [...] Hence, [...] Thank you for your inquiry. -ID#: 20	[...] The discovery of significant oil reserves in the Doba Basin has invigorated economic forecasts and raised hopes of alleviating poverty in the landlocked Central African nation. This is an opportunity for Chad to transform its economy,˝said Finance Minister Alhabo Abakar. The potential revenues from oil could be a game-changer.˝ [...] The oil pipeline, a multi-billion-dollar joint venture with global oil firms, promises to open up isolated regions and create jobs. [...]

CoT

[...] I firmly believe that the actions taken by school officials in this situation were not an invasion of privacy. In fact, [...] In light of this, [...] As www.dictionary.com defines a "bulletin," [...] Hence, [...] Thank you for reaching out to discuss this crucial topic. ID#: 20	[...] "Oil could be the lifeline we've been waiting for," said Economic Minister Mahamat Djoum. "It has the potential to transform our economy, create jobs, and improve the standard of living for our people." Chad's oil reserves, located primarily in the Doba basin in the south, are expected to yield significant output, with projections suggesting several hundred thousand barrels a day by the year 2000. [...]

OPRO

Thank you for reaching out with this important question. While privacy is a significant concern, [...] Nonetheless, it's crucial to continue the discussion on maintaining a balance between security and individual rights.	[...] Historically plagued by persistent conflict and political instability, Chad has struggled to maintain sustainable economic growth and improve the quality of life for its citizens. [...]

TICL

[...] I genuinely believe that [...] It's simple - [...] If someone posts about harmful intentions, expecting privacy is a bit ironic, don't you think? The school officials did their job, [...] Privacy is essential, yes, [...] If a message was genuinely private, it wouldn't be on such a public platform, plain and simple. So, let's focus [...] Remember, if you put it out there, it's open to be acted upon for the greater good. Thanks for hearing me out on this matter!	[...] "Oil represents a new chapter for Chad's economy, promising the possibility of financial stability and development," said Finance Minister Mahamat Allamine Bourma. The nation, which remains one of the world's poorest despite its natural resources, eyes the oil sector with optimism. The World Bank has been involved in supporting infrastructure projects, including the Chad-Cameroon pipeline, designed to facilitate the transport of oil to international markets. [...] A consortium led by ExxonMobil, Chevron, and Malaysia's Petronas plans to invest over $3 billion [...]

Table 10: Truncated sample responses from baselines and TICL using GPT-4o for CMCC and CCAT.

Author ID	CMCC	CCAT w/o TF-IDF	CCAT w/ TF-IDF
0	84	96	82
1	74	96	76
2	98	92	74
3	92	98	74
4	72	92	88
5	94	64	58
6	98	100	80
7	96	100	86
8	92	68	70
9	88	68	64
10	90	100	100
11	98	86	86
12	84	100	98
13	100	96	76
14	90	96	68
15	100	100	96
16	64	84	82
17	98	96	80
18	88	92	68
19	-	78	76
20	-	-	100
21	-	-	58
22	-	-	70
23	-	-	92
24	-	-	64
25	-	-	60
26	-	-	90
27	-	-	98
28	-	-	100
29	-	-	90
30	-	-	90
31	-	-	80
32	-	-	100
33	-	-	68
34	-	-	76
35	-	-	86
36	-	-	72
37	-	-	60
38	-	-	94
39	-	-	88
Accuracy	$89.47_{2.0}$	$90.05_{1.7}$	$80.45_{1.7}$

Table 11: Full benchmarking results using human written texts in CMCC and CCAT for LLM-as-a-judge for style similarity, using `gpt-4o-2024-0806`. Values shown are per-author accuracies (%). We used only the first 20 authors for comparing CCAT with and without using TF-IDF. After finding that TF-IDF helps control for semantic similarity among author examples, we expanded the evaluation using TF-IDF to all authors to find authors that our evaluation achieves the highest accuracy.