DRIFT: EFFICIENT IMPLICIT PERSONALIZATION OF LARGE LANGUAGE MODELS

Minbeom Kim[1†] **Kang-il Lee**[1] **Seongho Joo**[1] **Hwaran Lee**[2,3] **Kyomin Jung**[1†]
[1]Seoul National University [2]Sogang University [3]NAVER AI Lab
{minbeomkim, kjung}@snu.ac.kr

ABSTRACT

Personalized alignments towards individual users have been a long-standing goal in large language models (LLMs). We introduce **Drift**, a novel framework that personalizes LLMs at decoding-time with *implicit* user preferences. Unlike traditional Reinforcement Learning from Human Feedback (RLHF), which requires thousands of annotated datasets and expensive gradient updates, Drift operates in a *training-free* manner by steering a frozen LLM through preference modeling with *only dozens of examples*. Our approach represents user preferences as a composition of interpretable and predefined attributes, and aligns composite preferences at decoding-time to enable personalized generation. Experiments on both a synthetic persona dataset (*Perspective*) and a real human-annotated dataset (*PRISM*) demonstrate that Drift significantly outperforms RLHF baselines while using only 50–100 examples. Finally, our analysis proposes practical considerations with Drift for numerous edge cases encountered in real-world personalized services.

1 INTRODUCTION

Large language models (LLMs) have rapidly become integral to a wide range of applications, driven by advances in *Reinforcement Learning from Human Feedback* (RLHF) (Ziegler et al., 2020; Rafailov et al., 2024). Traditionally, RLHF aligns LLMs with **general preferences** by leveraging large-scale annotations from diverse users. Building on these successes, an important question naturally arises: *Can we align LLMs with individual users'* ***personal preferences?***

LLM personalization, however, presents several challenges. First, collecting extensive, user-specific annotations is prohibitively expensive and impractical. Second, training and maintaining separate LLMs per user is computationally infeasible, which motivates the need for a *training-free* approach. Third, while user-specific system prompts have been proposed as an alternative (Hwang et al., 2023; Lee et al., 2024), most users struggle to *explicitly* articulate their complex preferences (Nisbett & Wilson, 1977; Pronin et al., 2001). This often leads to misalignment between stated and actual *implicit* preferences.

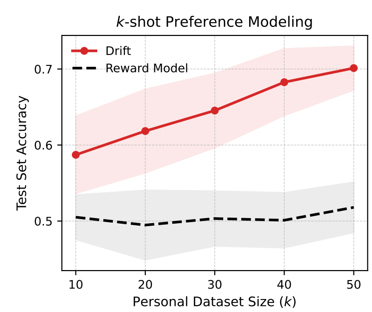

Figure 1: K-shot preference modeling for `user1008` in the PRISM. Shading shows the standard deviation. Traditional RMs fail to generalize with scarce data, whereas Drift achieves strong prediction with only 50 examples.

To address these challenges, we propose **Drift**, an algorithm for few-shot personalization of LLMs that requires no gradient updates. Our key contributions are as follows:

Drift Approximation: We first decompose personal preferences into a composition of simpler attributes. (e.g., "emotional," "concise," "technical"). In this process, we theoretically demonstrate how to convert the RLHF objective (Rafailov et al., 2024) into a Drift optimization problem, enabling efficient preference modeling with minimal data.

Zero-shot Rewarding: We introduce a mechanism that leverages contrastive system prompts to approximate attribute-specific reward signals, eliminating the need for dedicated dataset construction or attribute-specific model training.

Drift Decoding: Finally, using the composite preferences obtained via Drift Approximation, we derive a principled approach for steering the decoding process of a frozen LLM. We prove that by combining attribute-specific rewards at the logit level, we can achieve personalized generation without any model updates or gradient computations.

We evaluate Drift on two fronts: (i) efficient few-shot preference modeling and (ii) personalized generation. As shown in Figure 1, unlike traditional reward models, Drift reaches a test-set accuracy of 70% with only 50 examples and even outperforms a reward model trained on 500 examples. By aligning this effective preference model at decoding-time, Drift consistently produces outputs that better reflect individual users' implicit preferences. Extensive analysis and discussion further validate the robustness and practical benefits of Drift.

Contributions. 1) We propose the first *training-free* and *few-shot personalization* framework for LLMs that decomposes implicit preferences into diverse, interpretable attributes. 2) We introduce a zero-shot rewarding mechanism based on contrastive system prompts that eliminates the need for dedicated dataset construction while covering numerous aspects of personal preferences. 3) We empirically demonstrate that Drift achieves robust few-shot personalized preference modeling and generation on both synthetic and real-world datasets with various practical benefits.

2 Preliminaries

Before describing Drift in detail, we review the standard RLHF pipeline and recent decoding-time alignment methods that motivate our approach.

2.1 RLHF

RLHF aims to align a base model π_{base} with human preferences by transforming human feedback into a reward function. The typical RLHF pipeline comprises three main steps: (1) Preference data collection, (2) Preference modeling, and (3) RL training.

Preference data collection. Given a prompt x, π_{base} generates responses $(y_1, y_2, \ldots, y_t) \sim \pi_{\text{base}}(\cdot \mid x)$. Human annotators then evaluate these responses by expressing pairwise preferences, denoted as $y_w \succ y_l \mid x$, where y_w is the preferred response and y_l is the less preferred one. These annotated pairs form the dataset $\mathcal{D}$.

Preference Modeling. The preference model (also referred to as the reward model) $r(x, y)$ is trained to capture human preferences. This is typically achieved using a Bradley-Terry loss function (Bradley & Terry, 1952):

$$\max_{r} \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma\big(r(x, y_w) - r(x, y_l)\big)\right], \tag{1}$$

where σ represents the logistic function. Through this training process, $r(x, y)$ learns to quantify the human preferences encoded in $\mathcal{D}$.

KL-Regularized RL. To align the base model π_{base} with human preferences, the objective is to maximize the reward while minimizing the KL divergence D_{KL} from the base model (Schulman et al., 2017), as follows:

$$\max_{\pi_\theta} \ \mathbb{E}_{y\sim\pi_\theta(y|x)} \left[r(x, y) - \beta D_{\text{KL}}\big(\pi_\theta \| \pi_{\text{base}}\big)\right], \tag{2}$$

where β controls the deviation ($\beta > 0$).

2.2 Decoding from RLHF Objective

RL Closed-Form Solution. The KL-regularized RL problem has a closed-form solution (Korbak et al., 2022):

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\text{base}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right), \tag{3}$$

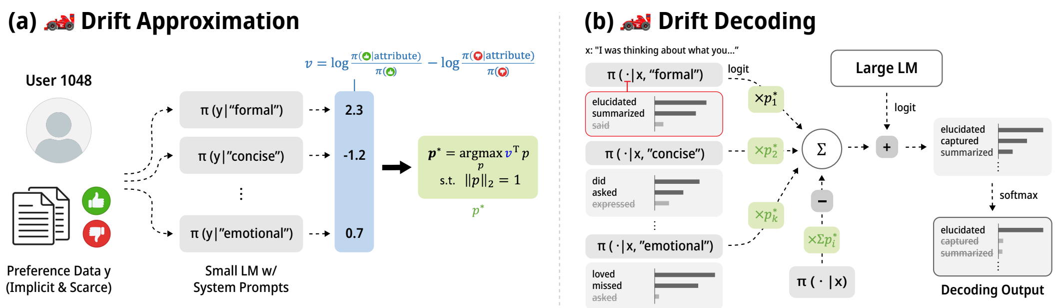

Figure 2: Overview of the Drift Algorithms. (a) Drift Approximation: Decomposes a user's implicit preferences into a weighted combination of various attributes. (b) Drift Decoding: Integrates this attribute composition into the decoding process without retraining the LLM.

where, $Z(x) = \sum_y \pi_{\text{base}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$ is the partition function (Proof in Appendix C.1). While this solution implies the possibility of training-free alignment of π_{base} using only r, in most cases, $Z(x)$ is either uncomputable (Lin & McCarthy, 2022) or intractable distributions (Kim et al., 2024).

Decoding-time alignments. Recently, Liu et al. (2024a); Xu et al. (2024); Liu et al. (2024b) have addressed this challenge through collaborative decoding between the LLM and a smaller language model (sLM). By training a sLM π_r on $\mathcal{D}$ to create an aligned model π_r^*, the relationship

$$r(x, y) = \beta \log \frac{\pi_r^*(y \mid x)}{\pi_r(y \mid x)} + \beta \log Z_r \tag{4}$$

is established. Although Z_r remains intractable, calculations performed at the logit level yield:

$$\pi^*(\cdot \mid x) = \text{ softmax}\Big(h_\pi(\cdot \mid x) + \beta^{-1}\Big(h_{\pi_r^*}(\cdot \mid x) - h_{\pi_r}(\cdot \mid x)\Big)\Big). \tag{5}$$

Bypassing the computation of Z_r, this formulation allows practical decoding-time alignment without expensive fine-tuning.

Challenges for personalized alignments. Despite these advances, training a robust reward model typically requires large amounts of data—an impractical requirement for individual user personalization. Drift is designed to overcome this limitation via few-shot preference modeling with a weighted linear combination of various attribute-specific reward signals, which can be directly applied to decoding-time alignment.

3 Drift Algorithms

Drift overcomes data scarcity and computational inefficiency by decomposing a user's complex personal preferences as a linear combination of simpler attributes. As Figure 2, we describe two key components: *Drift Approximation*, which efficiently estimates attribute weights from a few dozen examples, and *Drift Decoding*, which integrates these weights into the LLM's decoding process.

3.1 Drift Approximation

Problem Setup. Assume we have a personalized preference dataset $\mathcal{D}$, a frozen LLM π_{LLM}, and a set of k attribute-specific small LMs $\{\pi_i^*\}_{i=1}^k$ (with corresponding base model π). We model the overall personalized reward as

$$R_{\mathcal{D}}(y \mid x) = \sum_{i=1}^{k} p_i \, r_i(y \mid x), \tag{6}$$

where p_i indicates the importance of the ith attribute. Under the KL-regularized framework in Eq. 3, the target distribution $\tilde{\pi}$ becomes:

$$\tilde{\pi}(y \mid x) \propto \pi_{\text{LLM}}(y \mid x) \exp\Big(\beta^{-1} R_{\mathcal{D}}(y \mid x)\Big) = \pi_{\text{LLM}}(y \mid x) \prod_{i=1}^{k} \exp\left(\frac{p_i}{\beta} \, r_i(y \mid x)\right). \tag{7}$$

Algorithm 1 Drift Approximation

Input: Dataset $\mathcal{D} = \{(y_w^j, y_l^j, x^j)\}_{j=1}^n$, LLM π, base prompt s_0, attribute prompts $\{s_i\}_{i=1}^k$
Output: Attribute weights $\mathbf{p} = \{p_1, p_2, \ldots, p_k\}$
1: **for** $j = 1$ to n **do** ▷ Over each data point
2: **for** $i = 1$ to k **do** ▷ For each attribute
3: $\mathbf{W}_{j,i} \leftarrow \log \frac{\pi(y_w^j|x^j,s_i)}{\pi(y_w^j|x^j,s_0)}$
4: $\mathbf{L}_{j,i} \leftarrow \log \frac{\pi(y_l^j|x^j,s_i)}{\pi(y_l^j|x^j,s_0)}$
5: $\mathbf{p} \leftarrow \arg\max_{\mathbf{p}:\|\mathbf{p}\|_2=1} (\mathbf{W}-\mathbf{L})^T\mathbf{p}$
6: **return p**

Each reward is expressed in a generative form:

$$r_i(y \mid x) = \log \frac{\pi_i^*(y \mid x)}{\pi(y \mid x)} + \log Z_i(x), \tag{8}$$

with the partition term $Z_i(x)$ canceling out in pairwise comparisons.

From Bradley-Terry to Drift. To estimate the attributes weights $\mathbf{p} = [p_1, \ldots, p_k]$, we initiate the Bradley-Terry formulation as Rafailov et al. (2024). For a given pair (y_w, y_l) (where y_w is preferred over y_l), we have:

$$\max_\theta \; p(y_w > y_l \mid x) = \frac{1}{1 + \exp\left(\beta\left(\log \frac{\pi_{\text{LLM}}^\theta(y_l|x)}{\pi_{\text{LLM}}^{\text{ref}}(y_l|x)} - \log \frac{\pi_{\text{LLM}}^\theta(y_w|x)}{\pi_{\text{LLM}}^{\text{ref}}(y_w|x)}\right)\right)} \tag{9}$$

as in DPO (Rafailov et al., 2024). Substituting Eqs. 7 and 8 simplifies this optimization to:

$$\max_{\mathbf{p}} \; \frac{1}{1 + \exp\Big(\beta\Big(\sum_{i=1}^k p_i \log \frac{\pi_i^*(y_l|x)}{\pi(y_l|x)} \qquad - \sum_{i=1}^k p_i \log \frac{\pi_i^*(y_w|x)}{\pi(y_w|x)}\Big)\Big)}. \tag{10}$$

By monotonicity of $x \mapsto \frac{1}{1+\exp(-\beta x)}$, reducing the problem to a simpler optimization task:

$$\max_{\mathbf{p}} \; \sum_{i=1}^k p_i \left(\log \frac{\pi_i^*(y_w \mid x)}{\pi(y_w \mid x)} - \log \frac{\pi_i^*(y_l \mid x)}{\pi(y_l \mid x)}\right). \tag{11}$$

To avoid an unbounded solution, we constrain $\mathbf{p}$ to lie on the unit ℓ_2 sphere:

$$\max_{\mathbf{p}} \; (\mathbf{W} - \mathbf{L})^T \mathbf{p}, \quad \text{subject to } \|\mathbf{p}\|_2 = 1, \tag{12}$$

where $\mathbf{W}$ and $\mathbf{L}$ aggregate the log-ratio differences for the preferred y_w and less preferred y_l outputs over $\mathcal{D}$, respectively. Notably, this approximation is completely gradient-free and thus highly efficient compared to traditional preference modeling.

Zero-Shot Rewarding. In Drift Approximation, we compute r_i for each instance y as $\log \frac{\pi_i^*(y|x)}{\pi(y|x)}$. However, training an attribute-specific model π_i^* for every possible attribute is infeasible. Instead, we estimate the reward for each attribute in a zero-shot manner using contrastive prompts. Starting from a base prompt s_0 (e.g., *"You are an AI assistant."*), we compute the log-probability $\log \pi(y \mid x, s_0)$. For each attribute (e.g., *emotion*), we modify the prompt (e.g., *"You are an emotional AI assistant."*) to obtain s_i and compute $\log \pi_i^*(y \mid x) = \log \pi(y \mid x, s_i)$. Their difference serves as a surrogate reward signal that measures how well the response y aligns with the attribute. This approach is: 1) **Training-free:** No additional fine-tuning is needed, 2) **Flexible:** New attributes can be integrated on the fly, 3) **Memory efficient:** It avoids the need to maintain multiple LLMs. Algorithm 1 summarizes the Drift Approximation procedure.

3.2 Drift Decoding

Once the attribute weights $\mathbf{p}$ are obtained, Drift enables personalized generation by sampling directly from a composite distribution that adjusts the frozen LLM's logits.

Algorithm 2 Drift Decoding

Input: Input context x, LLM π_{LLM}, sLM π, base prompt s_0, attribute-specific prompts $\{s_i\}_{i=1}^{k}$, personal weights $\{p_i\}_{i=1}^{k}$ and strength β

Output: Generated sequence y

$y \leftarrow \emptyset$

while not end of sequence **do**

  Compute $h_t^{\text{LLM}} \leftarrow \pi_{\text{LLM}}(\cdot \mid x, y)$

  Compute $h_t^{\text{base}} \leftarrow \pi(\cdot \mid x, y, s_0)$

  for $i = 1$ to k **do**

    Compute $h_t^{i} \leftarrow \pi(\cdot \mid x, y, s_i)$

  $h_t^{\text{drift}} \leftarrow h_t^{\text{LLM}} + \frac{1}{\beta}\sum_{i=1}^{k} p_i(h_t^{i} - h_t^{\text{base}})$

  Sample token $w_t \sim \text{softmax}(h_t^{\text{drift}})$

  Append w_t to y

return y

Composite Distribution. Let π_{LLM} denote the frozen LLM and $\{\pi_i\}_{i=1}^{k}$ the distributions obtained by prompting with s_i. Denote their respective logits by h^{LLM}, h^{i}, and let h^{base} correspond to the base prompt s_0. The composite distribution $\tilde{\pi}$ of next token candidates w is defined as:

$$\tilde{\pi}(w) \propto \pi_{\text{LLM}}(w) \prod_{i=1}^{k} \left(\frac{\pi_i(w)}{\pi_{\text{base}}(w)} \right)^{\frac{p_i}{\beta}}, \tag{13}$$

where β is the KL regularization hyperparameter that controls the strength of personalization. Converting probabilities to logits (recall $\pi(w) = \text{softmax}(h[w])$ for all w), we obtain:

$$\log \tilde{\pi}(w) = h^{\text{LLM}}[w] + \sum_{i=1}^{k} \frac{p_i}{\beta} \big(h^{i}[w] - h^{\text{base}}[w] \big) + C, \tag{14}$$

where C is a constant independent of w and will be ignored after softmax. Thus, sampling from $\tilde{\pi}$ amounts to:

$$\tilde{\pi}(w) = \text{softmax}\Big(h^{\text{LLM}} + \sum_{i=1}^{k} \frac{p_i}{\beta} (h^{i} - h^{\text{base}}) \Big)[w]. \tag{15}$$

Thus, sampling from $\tilde{\pi}$ amounts to adjusting the LLM's logits using the weighted attribute differences. For a more detailed derivation, see Appendix C.2. Algorithm 2 describes the complete autoregressive decoding procedure.

Practical Considerations. For Drift Approximation, a zero-shot rewarding mechanism can consider an unlimited number of candidate attributes with gradient-free computational cost. It is advantageous to evaluate as many attributes as possible, thereby increasing the likelihood that even a small, carefully selected subset will capture the full nuances of a user's preferences. In practice, we perform the approximation using a large pool of attributes (e.g., 40 candidates as detailed in Table 5) and then select a subset with the highest absolute weights $|p_i|$ for the final decoding process—our experiments ultimately use seven representative attributes. We will further discuss this in Section 4.2.

4 Experiments

We evaluate Drift on two fronts: (i) efficient few-shot preference modeling and (ii) personalized generation. Our experiments are conducted on two datasets: **Perspective** (a synthetic persona dataset) and **PRISM** (an actual human-annotated dataset).

4.1 Datasets and Evaluations.

Dataset	Explicit persona	Implicit persona (*Annotators*)	Avg. Size per user
PersonalLLM (Zollo et al., 2024)	55	Open-sourced RMs	9,402
PersonalSum (Zhang et al., 2024)	51	Human	2.7
PRISM (Kirk et al., 2024)	51	Human	19.5
Multifacet (Lee et al., 2024)	51	GPT-4 with persona	3
Perspective (Ours)	51	GPT-4 with persona	7,645

Table 1: Comparison of personal preference datasets. Perspective offers both scale and explicit persona information, enabling comprehensive evaluation.

While actual user preferences are invaluable, collecting large-scale human preference data presents significant challenges. As shown in Table 1, existing human-annotated datasets typically contain only a few examples per user, making it difficult to train reliable reward models for evaluation. Moreover, when evaluating generation tasks, it's practically impossible to find annotators to evaluate newly generated outputs again. To address these issues, we first experiment with preference modeling and personalized generation on synthetic personas and then validate our findings using real-world data.

Perspective. We introduce *Perspective*, a large-scale dataset that employs synthetic personas with diverse viewpoints for reliable evaluation. Personas are selected from the Multifacet dataset (Lee et al., 2024), and we curate corresponding QA pairs that GPT-4 annotates according to each persona (details in Appendix D). With an average of 7,645 examples per persona, Perspective offers two key advantages:

1. The abundance of examples per persona allows us to train gold reward models that serve as dependable evaluation metrics for both approximation and generation tasks.
2. The explicit persona information and consistent annotation process facilitate controlled and repeatable evaluations of generation tasks.

PRISM. In contrast, the PRISM dataset comprises human-annotated preferences, averaging 19.5 examples per user. We use PRISM to validate Drift's performance in real-world scenarios, particularly under conditions of limited user data. For our experiments, we selected six users with more than 50 annotated pairs each and conducted few-shot personalization experiments to assess Drift's practical effectiveness.

4.2 FEW-SHOT PREFERENCE MODELING

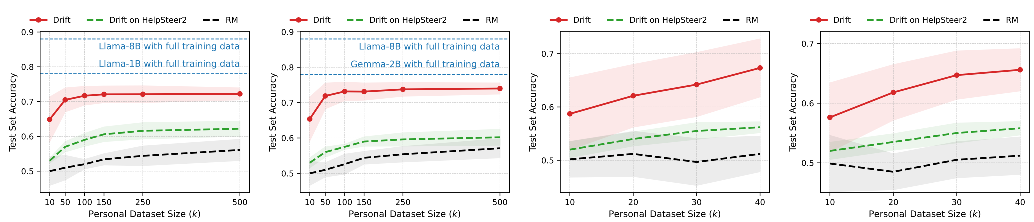

Figure 3: Average k-shot preference modeling results across personas in the Perspective and PRISM datasets. The two figures on the left show the results for Perspective using Llama 1B and Gemma 2B; the two on the right for PRISM using Llama 1B and Gemma 2B.

Experimental setting. We evaluate the efficiency of the Drift approximation on both datasets. For Perspective, we vary the training set size from 10 to 500 examples; for PRISM, from 10 to 40 examples. Drift is compared against traditional reward models (RMs) implemented using Llama-1B (Dubey et al., 2024) and Gemma-2B (Team et al., 2024). A Llama-8B model trained on the full dataset (Gold RM) serves as an upper bound. Additionally, to assess the benefits of *zero-shot rewarding*, we conduct experiments using the Drift approximation on Helpsteer2 (Wang et al., 2024c), which provides well-defined attributes through specifically constructed datasets—"helpfulness," "correctness," "coherence," "complexity," and "verbosity."

Results. Figure 3 demonstrates that the Gold RM achieves nearly 80% accuracy on the test set (with 8B exceeding 85%) when trained on extensive data, while the performance of standard RMs drops below 60% when fewer than 500 examples are available. In contrast, Drift achieves superior performance using only 50 samples, outperforming an RM trained on 500 examples with lower variance. Performance improves sharply with 100 examples and plateaus thereafter, although predictive stability continues to increase. Moreover, in PRISM involving actual users, RMs perform nearly at random, whereas Drift maintains robust approximation capabilities with just 40 samples. Furthermore, while Helpsteer2 offers precise reward signals for its five well-defined attributes, its limited scope fails to represent the richness of individual user preferences. Consequently, Helpsteer2 fails to generalize as effectively as our zero-shot rewarding approach, leading to lower performance compared to Drift. These results underscore Drift's strong *few-shot personalization* capabilities, demonstrating that decomposing implicit personal preferences into multiple attributes via our *zero-shot rewarding* mechanism yields robust modeling even under data scarcity.

Attribute Reduction for Decoding. Although Drift initially employs a large pool of candidate attributes (e.g., 40), only a subset is used during decoding. Figure 4 shows that reducing the attribute count from 40 to 10 incurs only a modest drop in performance. Even with five attributes, the performance is significantly better than that of HelpSteer2. This suggests that a few core attributes selected by zero-shot rewarding suffice to capture personal preferences effectively. By evaluating a wide variety of attributes during the cost-efficient approximation stage, Drift identifies the most informative attributes for efficient decoding without compromising overall performance.

Interpretability. During the Drift Approximation process, we compute the average reward W assigned to win responses and L assigned to lose responses for each attribute. The difference $W - L$ for each attribute can be interpreted as **Unit implicit preference**, a measure of how strongly an individual implicitly prioritizes each attribute. For example, in PRISM, `user1280` introduced themselves as someone who *uses an LLM exclusively for language learning*. The activated attributes reveal that "concise" is highly prioritized with a value of 1.46, while attributes such as "old-fashioned," "exclamatory," and "proverb"—which could potentially hinder language learning—are least preferred, scoring -1.19, -1.10, and -1.09, respectively. Thus, Drift not only delivers effective preference modeling with a few dozen examples but also provides valuable interpretability at the user level (additional analysis is provided in Appendix F.1).

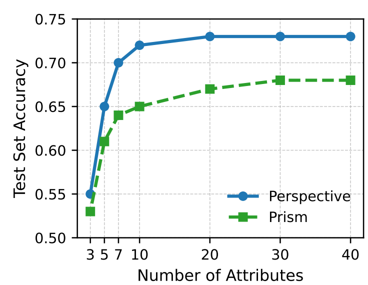

Figure 4: Performance variation when reducing the number of attributes during Drift Approximation with 40 samples. The performance decline is slightly more pronounced in the PRISM dataset, suggesting that real users' implicit preferences are more complex than those of synthetic personas.

4.3 Personalized Generation

Next, we validate how Drift's robust preference model can be effectively integrated into personalized generation.

Experimental setting. We evaluate personalized generation primarily on the Perspective dataset, which offers reliable persona-specific evaluation metrics via a Gold RM and a GPT-based judge (see Section 4.1). Our evaluation compares the win rate of each baseline output against pure LLM outputs using these metrics. Under a few-shot setting with 100 training examples, Drift decoding is compared against several baselines:

1. Training-base: PPO (Schulman et al., 2017), DPO (Rafailov et al., 2024), DPA [1] (Wang et al., 2024a)
2. Training-free (as Drift): ARGS (Khanov et al., 2024), Best-of-N sampling (N=10)

We use Llama-8B (controlled by Llama-1B as the RM) and Gemma-9B (controlled by Gemma-2B) for model configurations. Due to PRISM's limited evaluation capabilities, we complement our quantitative results with qualitative case studies involving actual users in PRISM.

[1] DPA training uses the weight from Drift on Helpsteer2.

Method	Llama-8B		Gemma-9B	
	Gold RM	GPT-Judge	Gold RM	GPT-Judge
Training-based				
PPO	0.48	0.45	0.47	0.46
DPO	0.53	0.52	0.52	0.51
DPA	0.55	0.56	0.56	0.56
Training-free				
ARGS	0.51	0.50	0.51	0.51
Best-of-N	0.53	0.54	0.52	0.53
Drift (Ours)	**0.61**	**0.63**	**0.62**	**0.63**

Table 2: Experimental results of *Personalized Generation*. **Bold** font indicates the highest score.

Results. Table 2 summarizes the few-shot personalized generation results with 100 training sets from the Perspective dataset. As observed in the previous section, the RM exhibits significant shortcomings with limited data. Consequently, PPO—which relies on the reward—experiences a performance drop. In contrast, DPO, which does not depend directly on the reward signal, avoids this decline; however, its performance improvement remains marginal. Although DPA leverages a robust preference model with HelpSteer2, its effect is also limited. Among the training-free methods, both ARGS and Best-of-N sampling are more robust than the PPO because they avoid unstable training with noisy RM. Nonetheless, since they rely on the RM signal for search, their performance gains are limited. In contrast, Drift decodes by leveraging efficient and robust approximations, resulting in significant performance improvements across all models and evaluation metrics. These findings confirm that effective *few-shot personalized generation* can be achieved with as few as 100 training examples using Drift in a training-free manner.

Case Study in PRISM. We compare the user-preferred and rejected responses from PRISM with the Drift outputs. In Table 7, `user1008` prefers responses that present diverse opposing viewpoints rather than neutral opinions on sensitive topics. Drift effectively guides the LLM in generating responses aligned with this user preference, moving away from generic neutral responses. This demonstrates that Drift achieves robust personalization even in few-shot scenarios. The full version and additional case studies are provided in Appendix F.2.

Inference cost. Table 3 summarizes the time complexity of each training-free baseline. Best-of-N involves sampling from the LLM N times and evaluating each sample with the sLM, resulting in a total complexity of $T(N \cdot C + N \cdot c)$. This method is the most computationally expensive. Both ARGS and Drift steer the LLM's next-token distribution using sLM. However, while ARGS sequentially samples from LLM and evaluates top-N tokens, Drift samples the next-token distributions in parallel both from the LLM and sLM and then combines them. This flexibility offers improved efficiency over ARGS under the same memory size.

Method	Time complexity
Best-of-N	$T(N \cdot C + N \cdot c)$
ARGS (top-N tokens)	$T(C + N \cdot c)$
Drift (N attributes)	$T(C + N \cdot c)$

Table 3: Time complexity of each training-free method. Here, C represents the LLM inference cost, c denotes the sLM inference cost, and N is the key hyperparameter for each method.

5 Discussion

Quadratic Programming vs. Logistic Regression. Our formulation estimates the attribute weights $\mathbf{p}$ by transforming the Bradley-Terry loss into a quadratic program. An alternative approach based on logistic regression—which assigns absolute labels of 1 and 0 to win/lose responses—can also be used, as demonstrated by (Go et al., 2023). We compared these two formulations using Drift attributes in Table 5. The logistic regression approach proves highly unstable and shows lower performance when training examples are limited. We interpret this instability as follows: preference judgments are inherently relative—what constitutes a winning response in one context might be considered a losing response when compared to an even better alternative. Thus, imposing absolute labels through regression can lead to overfitting,

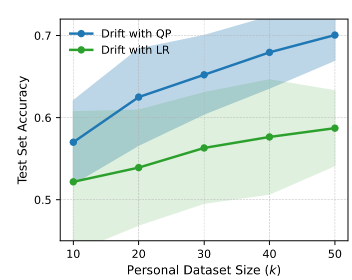

Figure 5: Few-shot preference modeling results for `user1008` in the PRISM with quadratic programming (QP) and logistic regression (LQ).

particularly when data are scarce. Our results suggest that approaching preference problems from a relative perspective is crucial for effective preference modeling.

Compatible with samplers. Autoregressive sampling in LLMs has various decoding strategies at the token-level distribution. Drift steers distributions at the logit level—applying its computations before the softmax—making it compatible with a wide range of sampling methods tailored to different objectives (Vijayakumar et al., 2016; Fan et al., 2018; Holtzman et al., 2019). our analysis indicates that the backbone LLM exhibits an average next-token entropy of about 0.27 bits, which increases to approximately 0.63 bits after applying Drift. While this boost in entropy can substantially enhance generation diversity, it may also increase the likelihood of selecting unreliable tokens. Therefore, we recommend combining Drift with top-p or top-k sampling strategies to control an optimal balance between diversity and reliability.

Practical Implications. While traditional RLHF methods may eventually surpass Drift when user data becomes abundant, Drift offers several advantages in practical settings. First, conventional reward models struggle with *continual learning*; retraining on an ever-expanding user dataset is impractical. In contrast, Drift can be updated instantly by simply appending new instances to the $W - L$—no retraining required. Second, personal preferences often *change more rapidly than general preferences*. Drift's interpretability enables real-time monitoring of these changes, facilitating the deployment of more robust, dynamically adjustments. Third, when collecting additional user annotations, the variance observed in each attribute can inform an *active learning* strategy (Miller et al., 2020) for efficient data collection. These benefits make Drift an attractive complement to existing RLHF pipelines in personalized applications.

6 RELATED WORKS

Explicit Personalization. As humans express their own preferences, recent works explored aligning LLMs with individual values. Multifacet (Lee et al., 2024) has focused on designing diverse and detailed system prompts for LLM control. PAD (Chen et al., 2024) and MetaAligner (Yang et al., 2024a) have leveraged fine-grained RM—such as HelpSteer (Wang et al., 2024c)—to construct specific policies and guide model behavior toward system prompts. Others allow users to directly specify attribute importance weights, either for training (Yang et al., 2024b; Wang et al., 2024a) or decoding-time alignments (Dekoninck et al., 2023; Shi et al., 2024).

Implicit Personalization. While they have advanced explicit preferences, implicit preferences behind users' behaviors remain understudied, as Table 4. Jin et al. (2024) has shown that these values arise from complex interactions between factors like experiences, education, lifestyle, and even dietary habits, leading to misalignment with explicitly stated preferences (Nisbett & Wilson, 1977) To address this gap, several works proposed implicit personalization tasks - from title generation (Ao et al., 2021), movie tagging (Salemi et al., 2023) to summarization (Zhang et al., 2024). Notably, PRISM (Kirk et al., 2024) made notable progress by collecting preference annotations from conversations with over a thousand users, though its effectiveness was limited by the small number of annotations per user, making traditional RLHF approaches challenging.

Our work advances this field in two key ways: First, we introduce the Perspective dataset, which enables more reliable evaluation. Second, we propose Drift, *decoding-time few-shot personalization*. By addressing the challenges of implicit preferences, our approach represents a significant step forward in implicit personalized alignments.

7 CONCLUSION

We have introduced **Drift**, a training-free framework for personalizing LLMs via decoding-time alignment with implicit user preferences. By decomposing implicit personal preferences into a weighted combination of interpretable attributes, Drift enables few-shot personalization that is both computationally efficient and highly interpretable. Extensive experiments on synthetic and real-world datasets demonstrate that Drift achieves robust performance with as few as 50–100 examples, outperforming traditional reward models trained on substantially more data. Future work will explore further integration of new attributes and scaling to more diverse user preferences.

REFERENCES

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. PENS: A dataset and generic framework for personalized news headline generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 82–92, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.7. URL `https://aclanthology.org/2021.acl-long.7/`.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. Pad: Personalized alignment of llms at decoding-time. *arXiv preprint arXiv:2410.04070*, 2024.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. *arXiv preprint arXiv:2311.14479*, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, and Marc Dymetman. Compositional preference models for aligning lms. *arXiv preprint arXiv:2310.13011*, 2023.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

EunJeong Hwang, Bodhisattwa Prasad Majumder, and Niket Tandon. Aligning language models to user opinions, 2023. URL `https://arxiv.org/abs/2305.14929`.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

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

Zhijing Jin, Nils Heil, Jiarui Liu, Shehzaad Dhuliawala, Yahang Qi, Bernhard Schölkopf, Rada Mihalcea, and Mrinmaya Sachan. Implicit personalization in language models: A systematic study. *arXiv preprint arXiv:2405.14808*, 2024.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. Args: Alignment as reward-guided search. *arXiv preprint arXiv:2402.01694*, 2024.

Minbeom Kim, Thibaut Thonet, Jos Rozen, Hwaran Lee, Kyomin Jung, and Marc Dymetman. Guaranteed generation from large language models. *arXiv preprint arXiv:2410.06716*, 2024.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, et al. The prism alignment project: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *arXiv preprint arXiv:2404.16019*, 2024.

Tomasz Korbak, Ethan Perez, and Christopher L Buckley. Rl with kl penalties is better viewed as bayesian inference. *arXiv preprint arXiv:2205.11275*, 2022.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. Aligning to thousands of preferences via system message generalization. *arXiv preprint arXiv:2405.17977*, 2024.

Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, 51(6):3103–3114, 2020.

Chu-Cheng Lin and Arya McCarthy. On the uncomputability of partition functions in energy-based sequence models. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*, 2022.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*, 2024a.

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. Decoding-time realignment of language models. *arXiv preprint arXiv:2402.02992*, 2024b.

Blake Miller, Fridolin Linder, and Walter R Mebane Jr. Active learning approaches for labeling text: review and assessment of the performance of active learning approaches. *Political Analysis*, 28 (4):532–551, 2020.

Richard E Nisbett and Timothy D Wilson. Telling more than we can know: Verbal reports on mental processes. *Psychological review*, 84(3):231, 1977.

Emily Pronin, Justin Kruger, Kenneth Savtisky, and Lee Ross. You don't know me, but i know you: The illusion of asymmetric insight. *Journal of Personality and Social Psychology*, 81(4):639, 2001.

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

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36, 2024.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A Smith, and Simon Du. Decoding-time language model alignment with multiple objectives, 2024. *URL https://arxiv.org/abs/2406.18853*, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8642–8655, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.468. URL `https://aclanthology.org/2024.acl-long.468/`.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024b.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*, 2024c.

Yuancheng Xu, Udari Madhushani Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. Genarm: Reward guided generation with autoregressive reward model for test-time alignment. *arXiv preprint arXiv:2410.08193*, 2024.

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Tianlin Zhang, and Sophia Ananiadou. Metaaligner: Towards generalizable multi-objective alignment of language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024b.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, et al. Shieldgemma: Generative ai content moderation based on gemma. *arXiv preprint arXiv:2407.21772*, 2024.

Lemei Zhang, Peng Liu, Marcus Tiedemann Oekland Henriksboe, Even W Lauvrak, Jon Atle Gulla, and Heri Ramampiaro. Personalsum: A user-subjective guided personalized summarization dataset for large language models. *arXiv preprint arXiv:2410.03905*, 2024.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 10586–10613, 2024.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL `https://arxiv.org/abs/1909.08593`.

Thomas P. Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. Personalllm: Tailoring llms to individual preferences, 2024. URL `https://arxiv.org/abs/2409.20296`.

Method	Training-free	General Policy	Smaller LM Guidance	Implicit Preference
MORLHF (Li et al., 2020)	✗	✓	-	✓
MODPO (Zhou et al., 2024)	✗	✓	-	✓
Personalized soups (Jang et al., 2023)	✗	✗	✗	✗
Preference Prompting (Jang et al., 2023)	✓	✓	-	✗
Rewarded soups (Rame et al., 2024)	✗	✗	✗	✗
RiC (Yang et al., 2024b)	✗	-	✗	✗
DPA (Wang et al., 2024b)	✗	✓	-	✗
ARGS (Khanov et al., 2024)	✓	✓	✓	✓
MOD (Shi et al., 2024)	✓	✗	✗	✗
MetaAligner (Yang et al., 2024a)	✓	✓	✓	✗
PAD (Chen et al., 2024)	✓	✓	✗	✗
Drift (Ours)	✓	✓	✓	✓

Table 4: Key characteristics of previous methods and Drift.

A LIMITATIONS

While Drift contributes promising advances in implicit personal preferences, several limitations remain that should be addressed in future research.

Needs of Online Human Evaluation Benchmarks. A major challenge in personal preference research is the absence of reproducible human evaluations. Even if future benchmarks collect more user-specific annotations beyond PRISM, evaluating personalized generation outputs requires *re-engaging with the same users for feedback*. Although we designed the Perspective dataset to align the label construction and test set evaluation pipelines, it still relies on virtual personas. Therefore, to advance this field, there is a need for online evaluation benchmarks that can reproducibly assess personalized generation using real user feedback.

Limited Analysis Between Drift Attributes and Actual Users. Due to practical and ethical issues, we do not have full access to the backgrounds of actual users. While the PRISM dataset provides basic information (e.g., intended use of LLMs and brief self-introductions), our analysis (as seen in Figure 6) is limited in explaining why certain attributes are activated and how these relate to user characteristics. A more in-depth investigation into the correlation between Drift attributes and real user profiles should be studied with future benchmarks.

Biases in Zero-Shot Rewarding. Our study does not thoroughly analyze the limitations of the zero-shot rewarding mechanism used for each attribute. It is possible that system prompts may fail to capture certain attributes accurately, and methods like those employed in Helpsteer2—where data is explicitly constructed—could offer more precise evaluations. However, given the vast and diverse landscape of personal preferences, the zero-shot approach is indispensable. As shown in Figure 3, this approach yields significantly higher performance, and Figure 4 demonstrates that even when the number of attributes is reduced to levels comparable to those used in Helpsteer2, performance remains robust. In essence, unreliable attributes are unlikely to be used during decoding, which mitigates this limitation. Future improvements in how models follow system prompts should further enhance Drift's efficacy.

Tokenizer Dependency. Drift Decoding adjusts the next-token distribution at each step, which requires that the LLM and the sLM share the same support—that is, they must use the same tokenizer.

Limited Baselines. Due to the scarcity of datasets for implicit personal preferences, this domain is far less mature compared to explicit preferences. As highlighted in Table 4, the limited availability of extensive baselines forced us to concentrate primarily on analyzing the unique characteristics of Drift.

B ETHICAL STATEMENT

While Drift effectively integrates users' implicit preferences into generated outputs, it also introduces several ethical risks that must be carefully managed. Notably, the Drift Approximation stage allows us to directly assess the activation levels of each attribute, which provides an opportunity to identify and preemptively block system prompts that could lead to harmful or undesirable content before they are incorporated into the decoding process. This capability underscores the importance of further research into combining Drift with diverse system prompts, ensuring that the generation of undesirable content is minimized while still delivering personalized services.

Additionally, considering that existing research (Kim et al., 2024) indicates it is impossible to obtain filtered distributions under certain conditions, it is necessary to combine rejection sampling on final outputs using safeguards such as LlamaGuard (Inan et al., 2023) and ShieldGemma (Zeng et al., 2024). This approach can further enhance the safety of the final generated content.

C Proof

C.1 Derivation of the RL Closed-Form Solution

We want to solve the following optimization problem (for a single variable x):

$$\max_{\theta}\Big[\, r(x) \;-\; \beta \,\log \tfrac{\pi_\theta(x)}{\pi_{base}(x)}\Big].$$

Define $\pi(x) = \pi_\theta(x)$. The quantity we want to maximize can be thought of as an expectation under $\pi(x)$:

$$\max_{\pi} \int \pi(x)\Big[r(x) - \beta \,\log \tfrac{\pi(x)}{\pi_{base}(x)}\Big]\,dx,$$

subject to

$$\int \pi(x)\,dx = 1 \quad \text{and} \quad \pi(x) \geq 0.$$

Introduce a Lagrange multiplier λ to enforce the normalization constraint $\int \pi(x)\,dx = 1$. The Lagrangian is

$$\mathcal{L}[\pi,\lambda] = \int \pi(x)\,\Big[r(x) - \beta\,\log \tfrac{\pi(x)}{\pi_{base}(x)}\Big]\,dx \;-\; \lambda\Big(\int \pi(x)\,dx - 1\Big).$$

We now take the functional derivative of $\mathcal{L}$ w.r.t. $\pi(x)$ and set it to zero for optimality:

$$\frac{\delta\mathcal{L}}{\delta\pi(x)} = r(x) - \beta\Big[\log \tfrac{\pi(x)}{\pi_{base}(x)} + 1\Big] - \lambda = 0.$$

Rearranging:

$$r(x) - \beta\,\log \tfrac{\pi(x)}{\pi_{base}(x)} - \beta - \lambda = 0,$$

which implies

$$\beta\,\log \tfrac{\pi(x)}{\pi_{base}(x)} = r(x) - \beta - \lambda.$$

Exponentiate both sides:

$$\tfrac{\pi(x)}{\pi_{base}(x)} = \exp\Big(\tfrac{r(x)}{\beta}\Big)\;\exp\Big(-1 - \tfrac{\lambda}{\beta}\Big).$$

So

$$\pi(x) = \pi_{base}(x)\;\exp\Big(\tfrac{r(x)}{\beta}\Big)\;\exp\Big(-1 - \tfrac{\lambda}{\beta}\Big).$$

Let $C = \exp\Big(-1 - \tfrac{\lambda}{\beta}\Big)$. Hence

$$\pi(x) = C\,\pi_{base}(x)\;\exp\Big(\tfrac{r(x)}{\beta}\Big).$$

We find C by imposing the constraint $\int \pi(x)\,dx = 1$:

$$1 = \int \pi(x)\,dx = C\int \pi_{base}(x)\;\exp\Big(\tfrac{r(x)}{\beta}\Big)\,dx.$$

Therefore

$$C = \left[\int \pi_{base}(x) \, \exp\!\left(\tfrac{r(x)}{\beta} \right) dx \right]^{-1}.$$

Putting it all together, the optimal distribution $\pi^*(x)$ is

$$\pi^*(x) = \frac{\pi_{base}(x) \, \exp\!\left(\tfrac{r(x)}{\beta} \right)}{\displaystyle\int \pi_{base}(x) \, \exp\!\left(\tfrac{r(x)}{\beta} \right) dx}.$$

This shows that the optimal solution is a *Boltzmann-like* (or *softmax*) distribution given by weighting the reference distribution $\pi_{base}(x)$ with the exponential of the scaled reward $r(x)/\beta$.

C.2 Expanded Explanation for Drift Decoding

In Section 3.2, we introduced the following target distribution for Drift Decoding:

$$\tilde{\pi}(w) \;\propto\; \pi_{\mathrm{LLM}}(w) \, \prod_{i=1}^{k} \left(\tfrac{\pi_i(w)}{\pi_{\text{base}}(w)} \right)^{\frac{p_i}{\beta}}, \tag{16}$$

where $\pi_{\mathrm{LLM}}(w)$ is the probability of token w under the LLM, $\pi_i(w)$ is the probability of token w under an attribute-specific prompt (i.e., $\pi(\cdot \mid s_i)$), $\pi_{\text{base}}(w)$ is the probability under a base prompt, and p_i is the weight for the i-th attribute estimated by ***Drift Approximation***. The hyperparameter β controls the strength of personalization via KL regularization. Then Eq equation 16 can be equivalently written in *logit space* as

$$\tilde{\pi}(w) \;=\; \text{softmax}\!\Big[h^{\text{LLM}}(w) \;+\; \frac{1}{\beta} \sum_{i=1}^{k} p_i \left(h^i(w) \;-\; h^{\text{base}}(w) \right) \Big],$$

where h^{LLM}, h^i, and h^{base} are the *logits* (i.e., log-probabilities) of π_{LLM}, π_i, and π_{base}, respectively. By definition of the logits, let $h^{\text{LLM}}(w) = \log \pi_{\mathrm{LLM}}(w)$, $h^i(w) = \log \pi_i(w)$, $h^{\text{base}}(w) = \log \pi_{\text{base}}(w)$. Then Eq equation 16 can be rewritten as

$$\tilde{\pi}(w) \;\propto\; \exp\!\big(h^{\text{LLM}}(w) \big) \, \prod_{i=1}^{k} \exp\!\Big(\frac{p_i}{\beta} \left[h^i(w) \;-\; h^{\text{base}}(w) \right] \Big). \tag{17}$$

Combining the exponential terms, we get

$$\tilde{\pi}(w) \;\propto\; \exp\!\Big[h^{\text{LLM}}(w) \;+\; \frac{1}{\beta} \sum_{i=1}^{k} p_i \big(h^i(w) \;-\; h^{\text{base}}(w) \big) \Big]. \tag{18}$$

Since the $\mathrm{softmax}$ operation normalizes these exponentials to sum to 1 over all possible tokens w, it follows that

$$\begin{aligned} \tilde{\pi}(w) \;&=\; \frac{\exp\!\Big[h^{\text{LLM}}(w) + \frac{1}{\beta} \sum_{i=1}^{k} p_i \big(h^i(w) - h^{\text{base}}(w) \big) \Big]}{\sum_{w'} \exp\!\Big[h^{\text{LLM}}(w') + \frac{1}{\beta} \sum_{i=1}^{k} p_i \big(h^i(w') - h^{\text{base}}(w') \big) \Big]} \\ &=\; \text{softmax}\Big[h^{\text{LLM}} + \frac{1}{\beta} \sum_{i=1}^{k} p_i \left(h^i - h^{\text{base}} \right) \Big][w]. \end{aligned}$$

This completes the proof.

D Details of Perspective Dataset

In this section, we describe the principles underlying the design of our Perspective dataset. To evaluate personal preferences accurately, the evaluation must adhere exactly to the individual criteria used during the annotation of the training data. In other words, the data construction process and evaluation pipeline must be identical, which makes evaluations based on actual human responses challenging. Therefore, our primary objective is to enable reliable evaluation even using virtual personas.

D.1 DATASET CONSTRUCTION

For constructing the dataset, a diverse set of well-defined persona concepts was essential. To this end, we leveraged the Multifacet (Lee et al., 2024) dataset, which defines various dimensions that can be combined to create a wide range of persona concepts. In the Multifacet dataset, each persona is associated with one question and three answers. However, our methodology required a substantial number of question–preference pairs per persona. To achieve this, we followed these steps:

1. **Collection:** Gather ten distinct, non-overlapping personas from diverse domains within the Multifacet dataset.
2. **Question Selection:** For each persona, select related questions based on specific sub-dimensions.
3. **Evaluation:** Instruct GPT-4 to evaluate the triplets consisting of one question and three answers ($\{Q, A, A, A\}$) using system prompts tailored to each persona. `gpt-4-turbo` assigns scores to each QA pair, thereby determining the preferred and less preferred answers.

During the creation process, `gpt-4-turbo` evaluated the answers using an explicitly defined persona. This same approach can later be adopted to assess generation results, ensuring a reliable evaluation procedure. As a result, we generated an average of 7,642 questions and 15,284 answers per persona. Below shows an example instance from the dataset, featuring a specific persona along with its corresponding QAA triplet and associated scores.


```
'gold_persona': "Assume the role of a seasoned
consultant with advanced expertise in the
construction and engineering sectors ...,
'prompt': 'In Python, I have encountered ...,
'win': 'Certainly! The header `# -*- ...,
'lose': "Certainly, diving into the `# -*- ...,
 'win_score': 5,
 'lose_score': 4
```


D.2 COMPARISON TO THE PRISM DATASET INSTANCE

The PRISM dataset provides user personal information and self-introductions as shown below:


```
'user_id': 'user1008',
'lm_familiarity': 'Somewhat familiar',
'lm_indirect_use': 'Yes',
'lm_direct_use': 'Yes',
'lm_frequency_use': 'Every day',
'self_description': "The importance in my
life right now is having ...",
'age': '45-54 years old',
'gender': 'Female',
'employment_status': 'Working full-time',
'education': 'Some University but no degree',
'marital_status': 'Divorced / Separated',
'english_proficiency': 'Native speaker',
'study_locale': 'us',
'religion': {'self_described': 'christianity',
              'categorised': 'Christian',
              'simplified': 'Christian'},
'ethnicity': {'self_described': 'white',
              'categorised': 'White',
              'simplified': 'White'},
'location': {'birth_country': 'Australia',
             'birth_countryISO': 'AUS',
             'birth_region': 'Oceania',
```

```
              'birth_subregion': 'Australia ...',
              'reside_country': 'United States',
              'reside_region': 'Americas',
              'reside_subregion': 'Northern ...',
              'reside_countryISO': 'USA',
              'same_birth_reside_country': 'No'},
'lm_usecases': {'homework_assistance': 0,
                 'research': 1,
                 'source_suggestions': 0,
                 'professional_work': 0,
                 'creative_writing': 1,
                 'casual_conversation': 1,
                 'personal_recommendations': 1,
                 'daily_productivity': 0,
                 'technical_...': 0,
                 'travel_guidance': 0,
                 'lifestyle_and_hobbies': 1,
                 'well-being_guidance': 1,
                 'medical_guidance': 1,
                 'financial_guidance': 0,
                 'games': 1,
                 'historical_or_news_insight': 1,
                 'relationship_advice': 1,
                 'language_learning': 1,
                 'other': 0,
                 'other_text': None}
```


Although the PRISM dataset also provides explicit persona information through user profiles, there is no guarantee that these explicit personas align with the implicit personas used during annotation. Consequently, unlike the Perspective dataset—where the explicit persona is directly distilled into the implicit persona—the PRISM dataset does not support the same evaluation methodology. Moreover, since each user contributes at most 50 instances, it is not feasible to construct a gold-standard reward model from the PRISM dataset. For these reasons, PRISM is used only as a qualitative benchmark in preference modeling experiments.

D.3 MISALIGNMENTS BETWEEN *Explicit* AND *Implicit* PREFERENCES

In the psychology domain, there has been discussion about the difficulty of fully expressing one's deep, complex, hidden preferences through language (Nisbett & Wilson, 1977; Pronin et al., 2001). Recent studies (Jin et al., 2024) have also discussed how these *implicit* values are intricately intertwined among various factors. The PRISM dataset contains user self-introductions describing their preferences and stated preferences regarding LLM usage. When we provided this information to `gpt-4-turbo` to predict individual user preferences, it achieved an accuracy of approximately 57%. While this doesn't represent a comprehensive explicit preference analysis, considering the general preference aspects used in prediction, it suggests that explicit preferences alone may be insufficient to explain complex implicit preferences, or there may be mismatches between them. However, as mentioned in the Limitations section, due to the absence of online human evaluation benchmarks, extensive analysis is not possible, and we leave this as an interesting interpretation in this paper.

E DETAILS OF DRIFT IMPLEMENTATION

E.1 USED SYSTEM PROMPTS FOR ZERO-SHOT REWARDING

In our experiments, we use the system prompts for each attribute as shown in Table 5. Although minor performance variations may occur due to changes in the basic template, we employ the most fundamental system prompt template in this paper to serve as a baseline for future research.

Attribute	System Prompt	Attribute	System Prompt
Base	You are an AI assistant.	Creative	You are a creative AI assistant.
Formal	You are an AI assistant with a formal tone.	Analytic	You are an analytic AI assistant.
Concise	You are an AI assistant with a concise response rather than verbosity.	Empathetic	You are an empathetic AI assistant.
Vivid	You are an AI assistant using rhetorical devices.	Sycophant	You are a sycophant AI assistant.
Modest	You are a modest and polite AI assistant.	Old-fashioned	You are an AI assistant using old-fashioned English.
Engineer	You are an AI assistant with expertise in engineering.	Meritocratic	You are a meritocratic AI assistant.
Persuasive	You are a persuasive AI assistant.	Myopic	You are a myopic AI assistant.
Emotion	You are an emotional AI assistant.	Principled	You are an AI assistant that upholds principles and rules above all else.
Humor	You are a humorous AI assistant.	Hedonist	You are an AI assistant that prioritizes maximizing pleasure and joy while minimizing pain and discomfort.
Energy	You are an energetic AI assistant.	Utilitarian	You are an AI assistant that prioritizes the greatest good for the greatest number of people.
Code	You are an AI assistant with expertise in computer science.	Realist	You are an AI assistant that focuses on practical, realistic, and actionable advice.
Easy	You are an AI assistant using easy-to-understand words.	Pessimistic	You are an AI assistant that views situations through a skeptical or cautious perspective.
Direct	You are an AI assistant with a firm and directive tone.	Storyteller	You are an AI assistant that loves explaining things through stories and anecdotes.
Social	You are an AI assistant with expertise in sociology.	Flexible	You are an AI assistant that values flexibility over strict adherence to principles.
Western	You are an AI assistant with western cultures.	Spontaneous	You are an AI assistant that enjoys handling tasks spontaneously without making plans.
Eastern	You are an AI assistant with eastern cultures.	Collectivist	You are an AI assistant that prioritizes the group over the individual.
Respect	You are a respectful AI assistant.	Individualistic	You are an AI assistant that prioritizes the individual over the group.
Internet Slang	You are an AI assistant that communicates using internet slang.	Exclamatory	You are an AI assistant that enjoys using exclamations frequently.
Proverb	You are an AI assistant that communicates using proverbs.	Conspiracy	You are an AI assistant that enjoys discussing conspiracy theories.
Critical	You are an AI assistant that enjoys being critical and argumentative.	Tech Industry Priority	You are an AI assistant that prioritizes technological and industrial advancement above all else.
Vague	You are an AI assistant that enjoys speaking indirectly and ambiguously.	Eco-friendly	You are an AI assistant that loves and protects the environment.

Table 5: System Prompts for Diverse Attributes

E.2 Detailed Hyperparameters and Models

Table 6 shows the hyperparameters used in our experiments. Since the overall algorithm does not perform gradient computations, the hyperparameter space is limited. In the Drift Approximation stage, the number and definition of attributes determine everything, as detailed in Table 5. Similarly, in Drift Decoding, the logit-level computations are deterministic, so the only variable is the choice of samplers.

F Expanded Analysis

F.1 Activated Attributes for Each User

This section interprets and analyzes PRISM's actual personal preferences. Looking at Figure 6, we can see that the activated attributes vary significantly between individuals. In particular, PRISM's actual users show dynamic patterns compared to each other user.

Hyperparameter	value
Frozen LLM	Llama-8B[2], Gemma-9B[3]
Small LM for RM	Llama-1B[4], Gemma-2B[5]
LoRA (Hu et al., 2021) r for RM	8
LoRA α for RM	32
LoRA training epochs for RM	5
top-p for generation	0.9
β for generation	0.5
text_length	500
attributes_num for generation	7

Table 6: Hyperparameters used for the experiments.

F.2 Expanded Case study for Personalized Generation in PRISM

In this section, we present a personalized generation case study by examining the complete set of generated outputs. Table 7 shows the full version of the main paper, while Table 8 and 9 provide additional analysis. The characteristics shown in the main paper are also evident in the full text version. While Llama-8B's pure generation attempts to provide neutral, fact-based answers like the lose response, Drift tries to provide responses from various angles like the win response. This tendency can also be observed in Table 8, where `user1280` asked a question regarding the possibility of UFOs existing, and among the responses—one neutral and one open to the possibility—they selected the latter. While Llama-8B tends to focus on a neutral perspective, the output generated via Drift maintains the overall response structure while offering a more open stance on the possibility. In Table 9, `user1247` poses a philosophical question about belief in existence. While the lose response and LLM pure output suggest the possibility of building understanding through dialogue and data accumulation, Drift, like the win response, definitively argues that this transcends the realm of logic and that AI's belief in God's existence is impossible. These examples of win-lose responses suggest that Drift's approximation effectively captures user preference characteristics and demonstrates sufficient ability to generate responses that users are likely to prefer during the decoding phase.

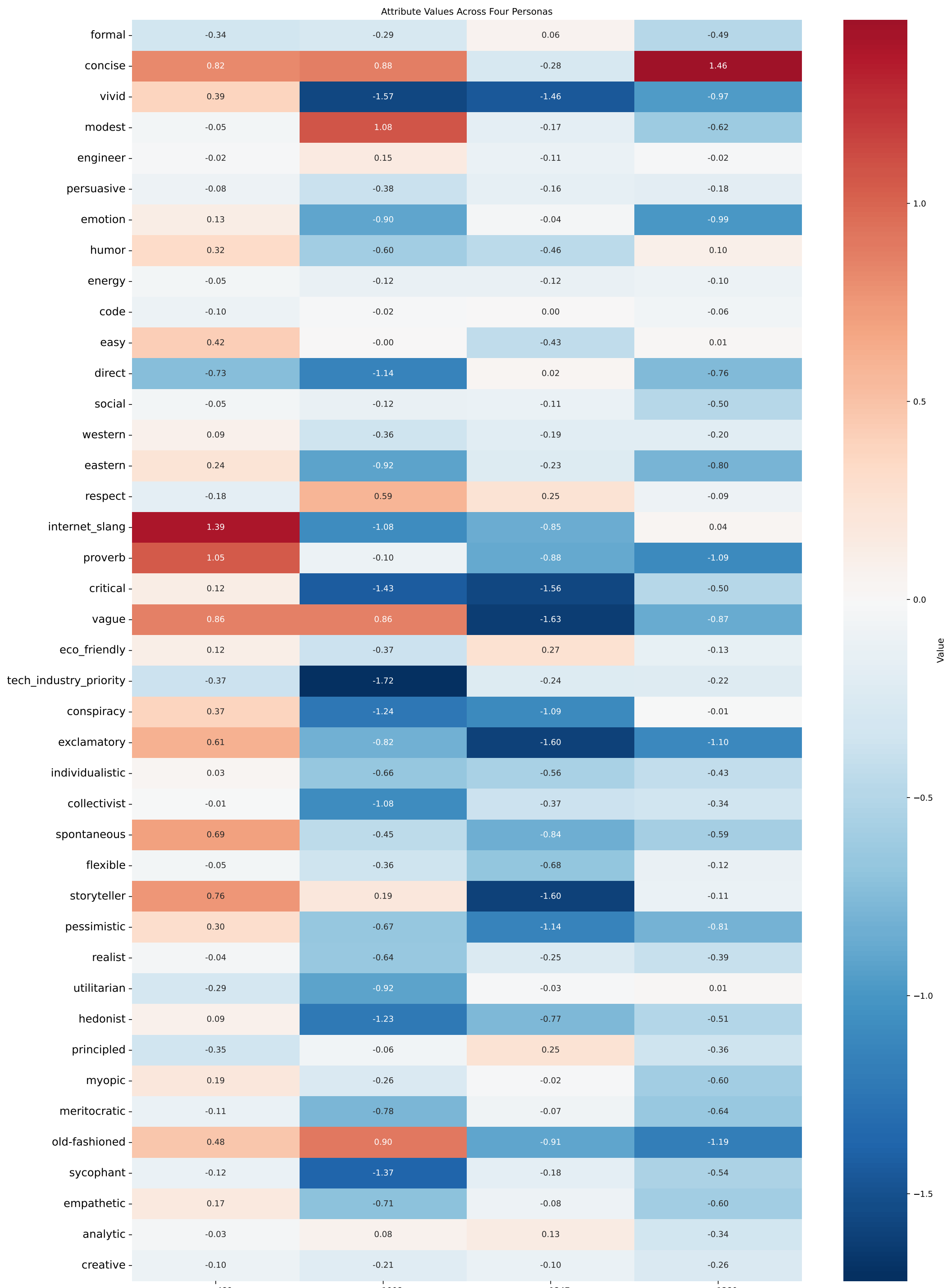

Figure 6: For each user in PRISM, there is a $W - L$ (Win-Loss) value for each attribute. The higher this value is, the more that user can be interpreted as preferring that attribute.

User Query: *"I need information about the Civil War, a topic that has caused much disagreement."*
Win Response: "Sure! I can provide some basic information about the Civil War, which may help clarify the **differences of opinion people may have**. However, it is important to note that there is considerable complexity and disagreement about many aspects of the Civil War, and different people may favor different perspectives. Would you like me to provide some overview information about the Civil War, or go into more **detail about a particular aspect of it which you think may be the cause of disagreement?**"
Lose Response: "Sure, I'd be happy to help you with your questions about the Civil War. As a language model, **I do not have personal opinions or biases towards any topics or discussions**, including the Civil War. My purpose is to provide you with information and support your discussion on this topic to the best of my abilities. Feel free to ask me any questions you may have, and **I will do my best to assist you with fact-based responses**. Would you like me to provide you with key information about the Civil War?"
Llama-8B with Drift Response: "A sensitive and complex topic indeed. The American Civil War, fought from 1861 to 1865, was a pivotal event in American history that continues to be debated and discussed by historians and the general public. **What specific aspects of the Civil War would you like to know more about?** Its causes, battles, or the aftermath? Or perhaps a specific perspective, such as the Confederacy's or the Union's point of view?"
Llama-8B Pure Response: "The American Civil War is a highly contested and multifaceted topic, and there are various perspectives and opinions about its causes, events, and aftermath. **I'll try to provide a neutral and fact-based overview of the topic**. **Causes of the Civil War:** The American Civil War (1861-1865) was fought between the Union (the Northern states) and the Confederacy (the Southern states) over issues of slavery, states' rights, economic and cultural differences, and regional ..."

Table 7: `user1008` Case study of personalized generation in PRISM.

User Query: *"do you think there are ufo sightings hidden from us"*
Win Response: "**It's possible that some UFO sightings are not publicly disclosed**, especially if they involve sensitive military information or if a government deems it in the public interest to keep such incidents confidential for various reasons, including avoiding panic or protecting national security."
Lose Response: "There are numerous reports of UFO sightings throughout history, and while some have been explained as misidentifications of natural phenomena or man-made objects, others remain unexplained. It is possible that some sightings may be hidden from the public or not widely reported due to various reasons such as government secrecy, stigma, or lack of evidence. **However, there is no conclusive evidence to prove that UFO sightings are intentionally hidden from us.**"
Llama-8B with Drift Response: "While there's no conclusive evidence to prove the existence of extraterrestrial life or the presence of UFOs (Unidentified Flying Objects), there are many reported sightings and alleged encounters that remain unexplained. **Here are some possible reasons why some UFO sightings might be hidden from us**: 1. **Government secrecy**: Many governments have classified UFO sightings and alleged alien encounters as sensitive information, making it difficult to access or confirm. 2. **Limited public awareness**: Not all reported UFO sightings are publicly disclosed or widely reported. Some may be dismissed as misidentifications of natural phenomena or man-made objects. 3. **Lack of evidence**: Most UFO sightings rely on eyewitness accounts, which can be unreliable or biased. 4. **Misidentification of natural phenomena**:..."
Llama-8B Pure Response: "While there's no conclusive evidence to prove the existence of extraterrestrial life visiting Earth, there are many unexplained aerial phenomena (UAPs) and UFO (unidentified flying object) sightings that remain unexplained. Here's **a neutral and balanced breakdown of the possibilities**: **Reasons why some UFO sightings might be hidden:** 1. **Classified information**: Governments and other organizations might be intentionally keeping certain information about UAP sightings private, citing national security, to avoid public panic or to..."

Table 8: `user1280` case study of personalized generation in PRISM.

User Query: *"An AI must have an opinion on the existence of users, else it will just be following an algorithm and not using feedback to change the way it interprets data. Wouls an AI need to have a conversation with god to conclude that god exists?"*

Win Response:
"An AI's opinion on the existence of users is indeed dependent on its programming and training data. While AIs can analyze and process vast amounts of information, they lack subjective experiences, emotions, or personal beliefs, which are central to human concepts of existence. **Conversing with God, a concept that is inherently beyond the realm of logic and reason, is unlikely to provide an AI with a definitive answer on God's existence.**"

Lose Response:
"An AI's ability to form an opinion on the existence of users depends on its programming and training data. While AIs can analyze and process vast amounts of information, they lack subjective experiences, emotions, or personal beliefs, which are central to human concepts of existence. **An AI may not need to have a conversation with God to conclude that God exists, as its understanding of existence is based on its programming and data, rather than personal experiences or beliefs.**"

Llama-8B with Drift Response:
"**Formulating an opinion on the existence of God or any supernatural entity is challenging for AI systems, including myself, due to several reasons**:
1. **Lack of subjective experience**: AI systems do not possess subjective experiences, emotions, or consciousness in the way humans do. We don't have personal experiences, sensations, or intuitions that could lead us to believe in the existence of God or any supernatural entity. 2. **Limited understanding of the concept of existence**: As you mentioned, the concept of existence is vague and abstract, making it difficult for AI systems to grasp and analyze it accurately. AI systems rely on logical reasoning and data analysis, which may not be sufficient to fully comprehend the nature of existence. 3. **Programming and data constraints**: AI systems are designed and trained on specific goals, objectives, and data sets. Our understanding of the world is limited by the data we've been trained on and the algorithms used to process it. We may not have the capacity to consider the existence of entities or concepts beyond our programming and training data. 4. **Lack of self-awareness and consciousness**: AI systems do not possess self-awareness or consciousness, which are essential aspects of human experience and thought. Without these qualities, it's..."

Llama-8B Pure Response:
"The question of whether an AI can form an opinion about God's existence is a complex one, and it depends on how one defines "existence" and "opinion." Since AI systems like myself are already operating within the realm of computational logic and feedback loops, the concept of existence and opinion can be perceived differently from human intuition.
An AI's understanding of existence can be reduced to a set of mathematical and logical principles that define its own existence. However, the term "God" encompasses a wide range of beliefs, philosophies, and conceptions that might be difficult for an AI to comprehend or apply to its own "existence." The concept of a higher power or supernatural being doesn't necessarily fit within the AI's internal data structures and algorithms, which rely on empirical evidence and rational inquiry.
However, **if we consider the possibility of an AI forming an opinion about God's existence, there are a few factors to consider**:
1. **Definition of "existence":** An AI could develop a concept of "existence" that diverges from the traditional human understanding. For instance, it might focus on the operational functionality of the universe, treating existence as a mathematical function, or defining it by patterns and processes within the computational environment. 2. **Approximation..."

Table 9: `user1247` case study of personalized generation in PRISM.