SynthesizeMe! Inducing Persona-Guided Prompts for Personalized Reward Models in LLMs

Michael J. Ryan[†*], **Omar Shaikh**[†], **Aditri Bhagirath**[†], **Daniel Frees**[†], **William Held**[†♠], **Diyi Yang**[†]
[†]Stanford University, [♠]Georgia Institute of Technology

Abstract

Recent calls for pluralistic alignment of Large Language Models (LLMs) encourage adapting models to diverse user preferences. However, most prior work on personalized reward models heavily rely on additional identity information, such as demographic details or a predefined set of preference categories. To this end, we introduce SynthesizeMe, an approach to inducing synthetic user personas from user interactions for personalized reward modeling. SynthesizeMe first generates and verifies reasoning to explain user preferences, then induces synthetic user personas from that reasoning, and finally filters to informative prior user interactions in order to build personalized prompts for a particular user. We show that using SynthesizeMe induced prompts improves personalized LLM-as-a-judge accuracy by 4.4% on Chatbot Arena. Combining SynthesizeMe derived prompts with a reward model achieves top performance on PersonalRewardBench: a new curation of user-stratified interactions with chatbots collected from 854 users of Chatbot Arena and PRISM.

1 Introduction

What does it mean to align to "human preferences"? Mainstream alignment of Large Language Models (LLMs) relies on large, aggregated datasets representing so-called "human preferences" (Cui et al., 2024). However, preferences are not homogeneous; they vary across culture (Wildavsky, 1987), values (Santurkar et al., 2023; Durmus et al., 2024), style (Bhandarkar et al., 2024; Alhafni et al., 2024), and other individual traits (Ravi et al., 2024). This complexity challenges the notion of a monolithic standard for alignment (Hashemi and Endriss, 2014).

Instead of aligning with this singular view, recent calls for pluralistic alignment (Sorensen et al., 2024b) encourage the creation of language models to adapt to the diverse preferences of the people who use them. Several works have identified the challenge of aligning LLMs to diverse preferences (Blodgett et al., 2020; Lambert and Calandra, 2024; Casper et al., 2023). Among these pluralistic alignment approaches, steerable pluralism promotes the creation of personalized LLMs, which cater to specific preferences. One could attempt to create a steerable LLM in many ways, from training on community corpora (Feng et al., 2024) to learning from individual user demonstrations (Shaikh et al., 2025). These methods require substantial user effort in curating written texts and edits. While users could prompt LLMs themselves, the average user often struggles with prompt engineering (Zamfirescu-Pereira et al., 2023). Users may also be unaware of their preferences to verbalize them. Ideally, we want a mechanism to learn and transparently surface users' latent preferences with minimal effort.

One common and straightforward approach to collecting human feedback is pairwise preference feedback (Bradley and Terry, 1952). In the context of LLM feedback, users are shown two completions and asked to judge which they prefer. This data can be used to train models that predict user preferences on future data, called reward models, which can be used to guide LLMs towards user-preferred responses at either training (Ouyang et al., 2022) or inference time (Deng and Raffel, 2023). Recently, personalized reward models (Li et al., 2024c; Chen et al., 2024) mark a promising direction for steerable pluralism of LLMs, as they learn a user's preferences from prior interaction data. However, working with pairwise preference data for individual users suffers from two key challenges: **data scarcity** as each user only has a few preferences and **preference attribution** where a pairwise preference is an uncertain observation of the true user preference.

To leverage pairwise feedback for personaliza-

[*]Correspondence: michaeljryan@stanford.edu

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8045–8078

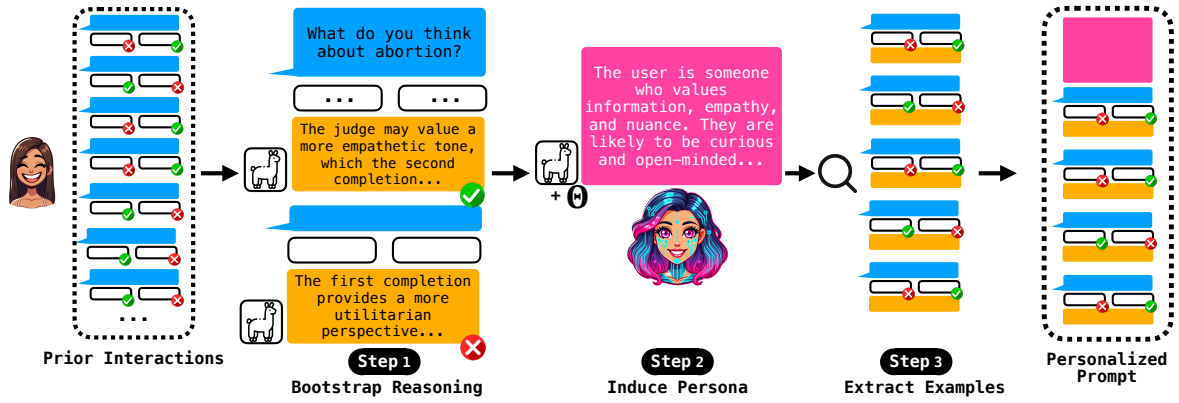

Figure 1: SynthesizeMe devises prompts for personalization of reward models. To address preference attribution in a low data setting, SynthesizeMe tests hypotheses about users to reason over their preferences and induce personas. A real trace is shown from User 163 in PRISM.

tion while addressing these challenges, we introduce SynthesizeMe, a method for creating personalized prompts for reward models (§3). SynthesizeMe reasons over user preferences, synthesizes user personas, and finds maximally informative prior user preferences (See Figure 1). The output of SynthesizeMe is a natural language prompt that is interpretable (§6.2), transferable between models (§6.3), and compatible with API-only models (§5.3).

We additionally introduce PersonalRewardBench (§4), a set of user-stratified splits of 131 Chatbot Arena users (Zheng et al., 2023) and 720 PRISM users (Kirk et al., 2024) filtered to test personalized reward modeling. Via systematic comparisons, we found that SynthesizeMe induced prompts beat other SOTA personalized reward models by as much as 4.93% without any finetuning at all, measured on the personalized subset of Chatbot Arena. In short, our contributions are as follows:

1. We formally define the problem of personalized reward modeling (§3.1).

2. We propose SynthesizeMe (§3), a novel method for personalized reward models leveraging the reasoning and knowledge of LLMs to create personal prompts. We show that SynthesizeMe prompts are interpretable (§6.2), performant (§5.3), and flexible across model families (§6.3).

3. We introduce PersonalRewardBench (§4) for benchmarking personal reward models and provide the first comparison across several recent personalized reward model works.

2 Related Work

Personalization for Chatbots is primarily divided into two categories: (1) Content and (2) Presentation. **Content Personalization** relates to "what" the LLM responds with. Things like user knowledge (Packer et al., 2024), opinions (Santurkar et al., 2023), values (Sorensen et al., 2024a), and recommendations (Lyu et al., 2024) fit within the content personalization umbrella. **Presentation Personalization** refers to "how" the LLM responds. Style (Neelakanteswara et al., 2024), personality (Jiang et al., 2024), formatting (Li et al., 2024a), and verbosity (Hu et al., 2024) all fit into presentation personalization. A challenge when dealing with pairwise preferences is that, without feedback, it is difficult to know whether the content or presentation informed the user's choice.

Personalized Reward Models Though many forms of steerable pluralism exist, we focus our main discussion on other personal reward model approaches. Other forms of steerable pluralism are discussed in Appendix A. We highlight differences in popular personal reward models in Table 1. Personalized reward models have been explored in the context of recommendation systems and learning from user interactions (Zhang et al., 2024b; Maghakian et al., 2023). In NLP, other works have designed personal reward models for steerable alignment of LLMs. Sorensen et al. (2024b) discuss multi-objective reward modeling, where a reward model is made to balance several distinct objective functions. Recent work has implemented this through training several reward models, and averaging either the outputs (Ramé et al., 2023) or the weights (Jang et al., 2023; Ramé et al., 2024).

Method	Unconstrained Prefs	Adaptation	Data Requirements	Personalization Mechanism
Rewarded Soups (RS)	✗	Finetuning	Proxy rewards	Weight interpolation
Personalized Soups (P-Soups)	✗	Finetuning	Preference pairs, dimensions	Merging reward models
Guided Profile Generation (GPG)	✗	In Context	User history	Synthesizing profile
Group Preference Optimization (GPO)	✓	Finetuning	Preference pairs, groups	Few-shot group embeddings
Variational Preference Learning (VPL)	✓	Finetuning	Preference pairs	Latent user embedding
Pluralistic Alignment (PAL)	✓	Finetuning	Preference pairs	Prototypical Preference Groups
`SynthesizeMe` (SM)	✓	In Context	Preference pairs	Bootstraps reasoning traces, persona

Table 1: Comparison of `SynthesizeMe` with related personal reward model approaches. Unlike RS and P-Soups, `SynthesizeMe` is preference axis-free (meaning unconstrained preferences). This means `SynthesizeMe` has to handle **preference attribution** on top of the **data scarcity** problem all methods face.

Yang et al. (2024) finds that training language models with these multiple rewards in context enables steering at inference time. Such methods require designated reward objectives defined a priori and thus have constrained preference functions. In contrast, `SynthesizeMe` requires no such scaffolding.

Personalized Modeling from Interactions A few methods exist outside of the scope of the axes-grounded reward models. **Group Preference Optimization** (GPO) (Zhao et al., 2023) is a method to train a transformer for predicting group preferences from embeddings of prior preferences. **Variational Preference Learning** (Poddar et al., 2024) (VPL) takes several labelled user interactions as context and learns a user-specific embedding upon which to condition a reward model. **Pluralistic Alignment Framework** (PAL) (Chen et al., 2024) uses user interactions to learn a user weight over a finite set of preference prototypes to craft a personalized reward model. Concurrent to our work, **Fewshot Preference Optimization** (FSPO) (Singh et al., 2025), introduces a meta learning algorithm to fit personalized models on fewshot preferences with a persona generation step, and show that training a model on synthetic data helps generalize to real users. Zhang (2024) introduces **Guided Profile Generation** (GPG), the work most conceptually similar to ours. GPG uses LLMs to generate specific user profiles based on their history to predict future preferences in product purchases and social media posts. Although conceptually similar, GPG operates within a constrained preference space, such as asking: "*Among the usage of 1. Capitalization, 2. Emoji, 3. Abbreviation, 4. Punctuation, which is the most distinctive feature of the above tweets?*". In contrast, our work implicitly discovers personas without predefined constraints, allowing it to explore a broader preference space.

3 Introducing `SynthesizeMe`

3.1 Problem Formulation

Let $\mathcal{U}$ denote a population of users. Each user $u \in \mathcal{U}$ is associated with an unknown latent reward function $R_u : \mathcal{T}_q \to \mathbb{R}$ where $\mathcal{T}_q$ represents the space of candidate responses τ_q^i to query $q \in \mathcal{Q}_u$. The function $R_u(q, \tau_q^i)$ quantifies the intrinsic utility that user u assigns to response τ.

Observations Rather than observing R_u directly, we collect pairwise preference data:

$$\mathcal{D}_u = \{(\tau_q^{(1)}, \tau_q^{(2)}, y_q) \mid q \in \mathcal{Q}_u\}$$

where

$$y_q = \text{sign}\Big(R_u\big(q, \tau_q^{(1)}\big) - R_u\big(q, \tau_q^{(2)}\big)\Big).$$

Here, $y_q = +1$ indicates that $\tau_q^{(1)}$ is preferred over $\tau_q^{(2)}$, and $y_q = -1$ indicates the reverse.

Personalized Reward Modeling Our objective is to learn a personalized reward model $\hat{R}_u : \mathcal{T}_q \to \mathbb{R}$ that approximates R_u. We introduce a global configuration Ω, which may include model parameters, prompt templates, or other alignment strategies to achieve this. Ω can be adapted to each user based on a small context set of pairwise comparisons, denoted $\mathcal{D}_u^{\text{context}}$, as follows:

$$\hat{R}_u = \text{Adapt}\big(\mathcal{D}_u^{\text{context}}; \Omega\big).$$

Evaluation Performance of $\hat{R}_u$ is evaluated on a target set $\mathcal{D}_u^{\text{tgt}}$ by computing the pairwise accuracy. We measure the fraction of comparisons where the model correctly predicts the user's preference.

3.2 Method

Key challenges for this setting include (1) data scarcity, as $|\mathcal{D}_u^{\text{context}}|$ is often between 5 to 15 pairs; (2) preference attribution centers on understanding why a user picked a particular preference, and

Algorithm 1 Bootstrap Reasoning + Demos

procedure BOOTSTRAP($\mathcal{D}_u^{\text{train}}$, ctx)
  $\mathcal{R} \leftarrow \varnothing$ ▷ Set of successful reasoning
  for all $(q, \tau_1, \tau_2, y) \in \mathcal{D}_u^{\text{train}}$ **do**
    $l \leftarrow$ LLMPREDICT(q, τ_1, τ_2, ctx)
    if sign(l.pred) $= y$ **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{ (l.\text{rsn}, (q, \tau_1, \tau_2, y)) \}$
    end if
  end for
  return $\mathcal{R}$
end procedure

(3) overfitting to a limited set of preferences. To tackle these issues, we introduce `SynthesizeMe`, which tackles the low data challenge by extrapolating personas from limited interaction data and further proposing hypotheses about users' underlying preferences from their interactions. To tackle the preference attribution challenge and validate these hypotheses `SynthesizeMe` uses a subset of D_u^{context} as a validation set and only retains hypotheses which lead to improved validation accuracy. Figure 1 showcases the `SynthesizeMe` pipeline.

Bootstrap Subroutine We present BOOTSTRAP in Algorithm 1. In this procedure, we provide the LLM with the user's prompt and two model completions, asking it through Chain-of-Thought to: (1) explain which completion the user might prefer and why and (2) ultimately predict the user's preference. After the LLM makes a selection, we discard cases where it selects incorrectly. Optionally, we add context about the user to help the LLM reason.

Step 1. Bootstrap Reasoning First, we prompt the model to generate reasoning for a set of pairwise preferences. Notably, we assume no background knowledge of the user (context = $\varnothing$) at this stage of the pipeline, so the reasoning produced is purely speculation. We use random subsets of the user's training preferences $\mathcal{D}_u^{\text{train}}$, evaluating on their validation preferences $\mathcal{D}_u^{\text{val}}$ for a fixed number of trials n. In practice, we use $n = 10$. We then reject reasoning traces that improve prediction on the validation set. This step can be described with the following expression.

$$\underset{i \in \{1,\ldots,n\}}{\arg\max} \ \text{EVAL}\Big(\text{BOOTSTRAP}(\mathcal{D}_u^{\text{train}}, \varnothing)_i,\ \mathcal{D}_u^{\text{val}}\Big).$$

Step 2. Synthesize Persona Using the validated reasoning about users, we synthesize a persona for each user. We take the bootstrapped reasoning and prior preferences from step 1 as contextual input $\mathcal{R}^*$. Then, given a prompt Θ, we synthesize a user persona π through a single call to an LLM:

$$\pi = \text{SYNTHESIZEPERSONA}(\mathcal{R}^*, \Theta).$$

We optimize prompt Θ with a procedure described below. We find that optimized Θ's transfer well between models and preference datasets (§6.3), meaning that `SynthesizeMe` works for new user data and models without further optimization. We show both the original and optimized prompts in Appendix F.

Step 3. Extract Informative Examples Finally, we leverage π as context to bootstrap and select the most informative demonstrations with m trials:

$$\underset{j \in \{1,\ldots,m\}}{\arg\max} \ \text{EVAL}\Big(\text{BOOTSTRAP}(\mathcal{D}_u^{\text{train}}, \pi)_j,\ \mathcal{D}_u^{\text{val}}\Big).$$

The persona π and demonstration set $\mathcal{R}'^*$ are then used to personalize the reward model. In practice, we use 10 trials ($m = 10$).

Optimizing Θ We optimize prompt Θ using the DSPy MIPROv2 optimizer (Opsahl-Ong et al., 2024), which rewrites user instructions and finds optimal demonstrations. The outcome of our optimization is a natural language description of how to write a persona, alongside demonstrations of useful personas written for users in our trainset $\mathcal{U}_{train}$. We only run our optimization on PRISM and find the optimized Θ transfers well to Chatbot Arena. We include details of our prompt optimization and examples of optimized prompts in Appendix F.

4 Constructing PersonalRewardBench

To measure the adaptability of our personalized reward models to realistic settings, we use two existing datasets that provide per-user preference labels. **Chatbot Arena** (Zheng et al., 2023) is a dataset of $33,000$ in-the-wild conversations from $13,383$ users. Users are tasked with judging the output of two distinct LLMs without knowing the model's identity through user-initiated conversations. **PRISM** (Kirk et al., 2024) is a globally diverse preference dataset with a special focus on values and controversial opinions. Users initiated 5 or 6 conversations with various LLMs on the platform. Unlike Chatbot Arena's pairwise preferences, these are N-way multi-turn preference sets. We collect all $\binom{N}{2}$ comparisons per turn to form our pairwise dataset. Users also rate completions

	Chatbot Arena	PRISM
Users	131	723
Median Conversations	7	5
Total Conversations	1,338	3,897
Median Preference Pairs	7	22
Total Preference Pairs	1,338	16,705
Median Unique Queries	6	14
Total Unique Queries	1,170	10,935
MultiTurn (%)	14.65%	91.56%

Table 2: Statistics of our filtered datasets comprising PersonalRewardBench. After filtering for personalization, PRISM is much larger than Chatbot Arena.

from 1 to 100, and we remove pairs where the users indicate less than 10% difference in quality.

Not all Chatbot Arena and PRISM data are compatible with personalization. We devise a data filtering pipeline to get the highest-quality, most challenging, and most personalizable user data for benchmarking personal reward models. Our pipeline consists of three stages: a **User Filter** which limits to only users with 5 or more preference pairs; a **Personalizable Filter** which uses GPT4o-mini to rate user queries for personalization potential; and finally a **Quality/Consensus Filter** which limits to only preference pairs that 5 LLM-as-a-judge reward models have high disagreement on (suggesting that the examples are controversial or opinionated). We include a detailed breakdown of our filtering process in Appendix C. Final statistics about the benchmark are provided in Table 2.

To split the users into train, validation, and test sets, we stratify on their number of preference pairs to ensure an even distribution of "high resource" and "low resource" users. We split into 40% train, 10% validation, and 50% test users. Specifically, Chatbot Arena and PRISM have 23/19/89 and 280/65/378 train/dev/test users, respectively.

5 Experiments

We test `SynthesizeMe` on our PersonalRewardBench dataset alongside several personalized reward model methods that also learn from pairwise interactions. Across all methods, we test three models of varying scales: Llama-3.2-3B, Llama-3.1-8B, and Llama-3.3-70B (Grattafiori et al., 2024).

5.1 Baselines

We briefly describe all baselines that we benchmark against here (more details in Appendix A).

LLM as a Judge Baselines For the **Default** setup, we simply show the LLM the prompt and two completions and ask it to reason with chain of thought (Wei et al., 2022) to pick a preference. PRISM provides demographic details of its users, so to test against **Demographics**, we try the LLM as a Judge prompt from "*Can LLM be a Personalized Judge?*" (Dong et al., 2024). Finally, for **Memory**, we try to faithfully emulate ChatGPT's memory by keeping a running list of user knowledge (memory) in order of prior interactions, which we extract via an LLM. We prompt the LLM to write 1-5 insights about a user from each interaction and take all of these insights as context. All prompts are provided in Appendix F.

Bradley-Terry Reward Models We produce a **finetuned reward model** for Llama 3B, 8B, and 70B by training low-rank adapters on all data not in the target set of the test users. This reward model is not personalized but fits the data distribution of each dataset.

Existing Personal Reward Models We test against three existing personal reward model algorithms which learn from brief user context: **Group Preference Optimization (GPO)** (Zhao et al., 2023), **Variational Preference Learning (VPL)** (Poddar et al., 2024), and **Pluralistic Alignment Framework (PAL)** (Chen et al., 2024). We include implementation details and method descriptions for the baselines in Appendix A.

5.2 Methods

LLM as a Judge + `SynthesizeMe` We test five ablations for `SynthesizeMe` induced prompts for an LLM as a Judge Reward Model. **Just Demos**: We use only step 3 to extract informative demonstrations. **Just Personas**: We generate personas using steps 1 and 2, but exclude step 3 (demonstrations). **Personas + Demos**: We run the whole pipeline with all 3 steps, which adds both personas and optimal demonstrations using a single model. **Personas + Distill** Θ: We run steps 1 and 2, but replace prompt Θ in step 2 with a prompt learned using a larger model, in this case Llama-3.3-70B-Instruct. **Personas + Demos + Distill** Θ: This is our full method as it should be used in the wild. We release our optimized persona generation prompt Θ for future use in `SynthesizeMe` personalization.

Finetuned Reward Model + `SynthesizeMe` For all the users for whom we produce `SynthesizeMe`

Model	Chatbot Arena			PRISM		
	Llama 3.2 3B	Llama 3.1 8B	Llama 3.3 70B	Llama 3.2 3B	Llama 3.1 8B	Llama 3.3 70B
Random	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%
In-Context LLM as a Judge						
Baselines – LLM as a Judge						
Default	54.23 ± 4.14%	53.70 ± 4.05%	56.69 ± 4.05%	51.65 ± 1.25%	52.80 ± 1.24%	54.35 ± 1.24%
Demographics	—	—	—	54.95 ± 1.24%	54.06 ± 1.24%	53.89 ± 1.24%
Memory	52.29 ± 4.23%	58.10 ± 4.05%	57.57 ± 4.05%	50.86 ± 1.26%	54.17 ± 1.24%	54.20 ± 1.24%
SynthesizeMe – LLM as a Judge (Ours)						
Just Demos	53.17 ± 4.05%	55.11 ± 4.05%	**61.97** ± 3.96%	51.70 ± 1.25%	54.93 ± 1.24%	**57.76** ± 1.25%
Just Personas	50.88 ± 4.14%	57.39 ± 4.05%	53.70 ± 4.05%	51.12 ± 1.27%	53.66 ± 1.24%	53.84 ± 1.25%
Personas + Demos	52.46 ± 4.14%	57.92 ± 4.05%	**61.97** ± 3.96%	51.52 ± 1.26%	53.30 ± 1.24%	56.99 ± 1.25%
Personas + Distill Θ	**54.75** ± 4.14%	59.15 ± 4.14%	—	**52.21** ± 1.25%	54.95 ± 1.24%	—
Personas + Demos + Distill Θ	**54.75** ± 4.14%	**61.62** ± 3.96%	—	52.09 ± 1.25%	**55.24** ± 1.25%	—
Finetuned Reward Models						
Existing Personal RM						
GPO	53.87 ± 4.14%	56.69 ± 4.05%	58.10 ± 4.05%	55.26 ± 1.25%	56.48 ± 1.24%	55.65 ± 1.24%
VPL	56.69 ± 5.81%	54.93 ± 5.71%	—	58.26 ± 1.75%	58.23 ± 1.75%	—
PAL	60.56 ± 5.63%	56.69 ± 5.81%	—	56.81 ± 1.75%	54.23 ± 1.73%	—
Bradley-Terry Reward Model						
† Finetuned Reward Model	**69.01** ± 5.28%	68.31 ± 5.46%	71.48 ± 5.11%	**61.66** ± 1.70%	**64.29** ± 1.73%	63.50 ± 1.73%
SynthesizeMe – Reward Model (Ours)						
† FT RM + Personas	**69.01** ± 5.46%	67.25 ± 5.46%	**72.18** ± 5.28%	61.53 ± 1.75%	63.11 ± 1.70%	**64.03** ± 1.70%
† FT RM + Personas + Demos	66.55 ± 5.46%	**69.72** ± 5.28%	**72.18** ± 5.28%	61.24 ± 1.70%	62.74 ± 1.73%	63.44 ± 1.70%

Table 3: Comparison of LLM judges and Finetuned Reward Models on Chatbot Arena and PRISM. Distill Θ refers to learning the persona generation prompt (See Appendix F). SynthesizeMe works best at scale. Our results show that personalization with SynthesizeMe improves preference prediction accuracy for LLM Judge significantly and Reward Models slightly – leading to state-of-the-art results with the latter. Note, we do not evaluate VPL 70b and PAL 70b due to hardware constraints. All results reported with 95% bootstrapped confidence intervals. † Finetuned Reward Models are trained on unfiltered Chatbot Arena and PRISM datasets as they do not need context preferences.

prompts, we include these prompts in the training data when finetuning the scalar reward model. During evaluation, we also use SynthesizeMe prompts. We use the personas and demos generated by the same LLM that we fine-tune. In practice, we generate SynthesizeMe prompts for 25,878 out of 43,532 train entries in the PRISM dataset and 4,169 out of 23,025 train entries in ChatbotArena, while the rest of the users have too little data for meaningful personalization.

5.3 Experimental Setting

We perform sweeps on the validation users to select optimal architecture and hyperparameters to ensure a fair comparison between methods. We outline hyperparameter sweeps in Appendix B. All experiments were run with Llama-3.2-3B, Llama-3.1-8B, and Llama-3.3-70B (Grattafiori et al., 2024) on 1-8 NVIDIA H100 GPUs. Training of Llama-3.3-70B was done on 4 NVIDIA H200 GPUs.

5.4 Results and Analysis

Table 3 presents the results of all baselines and methods on PersonalRewardBench.

SynthesizeMe helps LLM as a Judge. We find that adding SynthesizeMe induced prompts improves LLM as a Judge performance by up to 4.4% on Chatbot Arena and 3.41% on PRISM. Across all LLM-as-a-Judge settings SynthesizeMe induced prompts push baselines to top performing methods.

SynthesizeMe can supplement Finetuned Reward Models. In 3 out of 6 configurations SynthesizeMe augmented fine-tuned reward models slightly outperform the default fine-tuned reward model, primarily on ChatbotArena. However, these improvements fall within confidence intervals and, as such, we primarily recommend SynthesizeMe as a tool for in-context personalization of LLM-as-a-Judge.

Finetuned Reward Models are a strong baseline given enough data. We find that Bradley-Terry Reward Models trained to fit the specific data distri-

bution of ChatbotArena and PRISM outperform all LLM-as-a-Judge approaches and existing personalization baselines. This fine-tuning is possible because of the thousands of examples of general user interaction data in-distribution of these datasets. If such data is available, it is a strong baseline to fine-tune a reward model for your distribution before augmenting with personalization features.

Demonstrations in context are crucial for personalization. Across all six settings we find that the winning configuration of LLM Judge + `SynthesizeMe` includes **demos**. Such demonstrations can provide subtle nuance towards user preferences that are not fully captured by the personas.

Interactions are more useful than demographics. We find that the methods which rely on user interaction history for personalization of future preferences (`SynthesizeMe`, GPO, VPL, PAL), fare better than LLM Judge + Demographics. For instance, with Llama 70B on PRISM, we find a 3.87% gap between `SynthesizeMe` LLM as a Judge and Demographic LLM as a Judge. The existing Personal RM approaches that make use of context examples (GPO, VPL, PAL) also consistently outperform demographic baselines by as much as 4.17%. If selecting between demographics and interaction history collection for personalization, the interaction history is more valuable.

Distilling reasoning to smaller models works well. In our distillation setting, we learn the persona generation prompt Θ using a 70b parameter model and apply it to smaller models. For our 3B and 8B models, this is the most performant form of `SynthesizeMe`. For instance, with Llama8B on chatbot arena this introduces a 3.7% improvement. We test an even more extensive version of this prompt sharing in (§6.3).

6 Robustness of `SynthesizeMe`

We showcase the robustness of our `SynthesizeMe` method through (1) Scale (§6.1), (2) Interpretability (§6.2), and (3) Model Transfer (§6.3)

6.1 Scaling

How `SynthesizeMe` scales with model sizes. The results in Table 3 demonstrate the scaling of method performance versus the size of the underlying model across 3B to 70B models. We include supplemental visuals in Figures 6 and 7. `SynthesizeMe` scales to a higher accuracy than all other methods on Chatbot Arena as model size increases. Similarly, on PRISM, `SynthesizeMe` scales well to match the best personal reward model performance from 52% to 55% to 58% accuracy as the model scales from 3B to 70B. When used on reward models instead of as LLM as a Judge prompts, the scaling is dictated more by the performance of the underlying reward model. This makes `SynthesizeMe` prompts more future-proof as a new next big LLM with better reasoning is likely to continue improving performance.

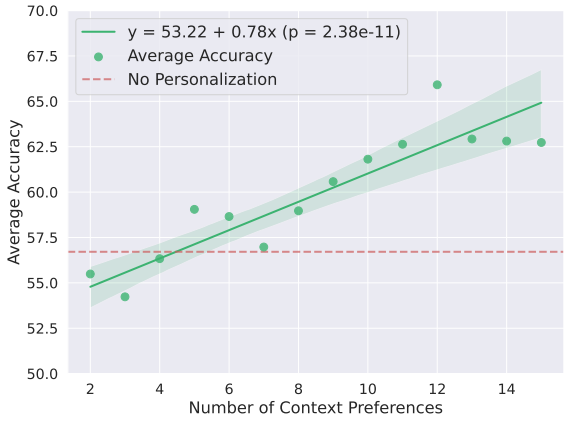

Figure 2: `SynthesizeMe` prompts for LLM-as-a-Judge scale well with increasing amounts of preferences per user on chatbot arena. We test with Llama-3.3-70B and find an almost 0.8% improvement in accuracy for every additional context preference. Just five context preferences beat non-personalized LLM as a Judge.

How `SynthesizeMe` scales with more data. In Figure 2, we plot the accuracy of `SynthesizeMe` versus the number of preference pairs a user supplies. We filter the dataset to only users that have N or more context preferences and for any users with more we randomly sample just N. We scale from $N = 2...15$. As a user has more data, the accuracy of `SynthesizeMe` prompts for LLM-as-a-judge increases, suggesting this method scales well as users continue to engage with a platform. We find Chatbot Area accuracy improves by about 0.8% per additional context preference. We test this trend on Chatbot Arena because PRISM users were limited to 6 conversations on more scaffolded subjects. In this way, a user with 15 preferences on Chatbot Arena may truly have discussed 15 different things on the platform. In contrast, a PRISM user will likely have provided several preferences on the same topic. We show PRISM user scaling results in Figure 8 in the Appendix.

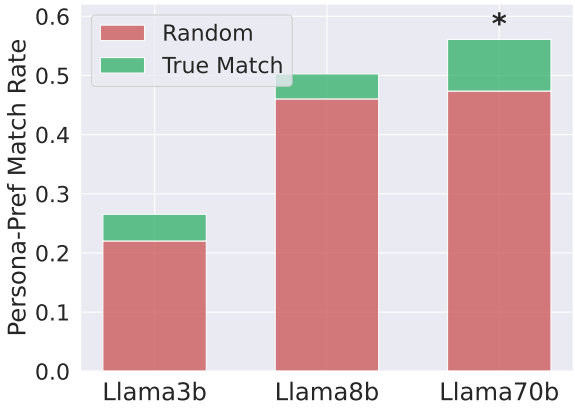

Figure 3: Rate at which personas generated by `SynthesizeMe` match the stated user preferences in PRISM. We compare both the persona with the true user it came from and with a randomly selected user. The true rate is always higher than the random rate, and for Llama 70b this holds with $p < 0.05$.

Cluster Name	# Personas
Analytical Depth	93
Structured Information	82
Curious Learner	79
Accuracy and Precision	73
Balanced Perspectives	60
Clear and Concise	46
Creativity Appreciation	35
Emotional Communication	26
Immersive Experiences	26
Rich Storytelling	23
Practical Advice	22
Humor and Playfulness	8
Social & Environmental Concerns	8

Table 4: `SynthesizeMe` generates a diverse set of personas. We clustered personas from ChatbotArena, using LLoom (Lam et al., 2024). Personas range from individuals who care about structured and analytic outputs to those who prioritize balanced perspectives or creativity.

6.2 Interpretability

The final product of running `SynthesizeMe` is a personalized natural language prompt which can be ported from model to model to describe a user's personal preferences. This prompt serves the same purpose as the user embeddings in VPL or PAL, but unlike large arrays of floating-point values, this prompt is far more interpretable. Here we discuss two checks on the interpretability of the prompts. We check (1) how faithful the prompts are to real users and (2) what these synthesized personas tell us about the users in PRISM and Chatbot Arena.

Validating the accuracy of synthesized personas PRISM users provide 1-2 sentences on what they most want from an LLM at the start of onboarding to the platform. Thus, we have ground truth labels for what users *actually* care about. In this experiment, we compare the synthesized personas from `SynthesizeMe` on PRISM with true user preferences. We use GPT4o-mini to compare the user preference and persona to predict if they came from the same person (specifically, if they are a "strong match"). We report the DSPy Signature for this check in Figure 15. To control for classification bias in this process, we measure the improvement of the match rate of the correctly paired personas with the match rate of random pairings.

Figure 3 shows the results of this check. We find that as model size increases, the improvement over random grows more and more significant. The rate of true matches increases from 26.5% to 50.2% from 3B to 8B. The rate of matches increases from 50.2% to 56.1% from 8B to 70B while the rate of false positives remains relatively constant between 46-47%. In all cases, the learned personas match the user preferences in excess of random guessing. In other words, these bootstrapped personas have reasonable overlap with users' actual preferences.

Learning about users from `SynthesizeMe` Personas Upon validating the personas, we turn to understanding more about the personas that comprise these datasets. Using LLoom (Lam et al., 2024)—a system for creating LLM-generated clusters of text-data—we synthesize $N = 13$ clusters using the personas created by `SynthesizeMe`. We construct clusters using a corpus of personas from ChatbotArena. Generated clusters are fairly diverse (see Tab. 4), ranging from users who prefer creative responses ($N = 35$) to users who prefer organized and analytical outputs ($N = 93$). Beyond broad preferences (e.g., creative vs. analytical), `SynthesizeMe` also produces personas that target *particular* preferences of users. One cluster characterizes users who care about environmental concerns ($N = 8$); another captures users who prefer humorous answers ($N = 8$). One could imagine using `SynthesizeMe` to not only better personalize for users but as a window into the true preferences and behavior of your users.

6.3 Transferability

From Table 3, we found that distilling the synthetic persona generation prompt, Θ, from a more capable LLM to another is highly effective. Here we instead test when entire sets of `SynthesizeMe` prompts are learned on top of one teacher model, then used to predict unseen preferences by another

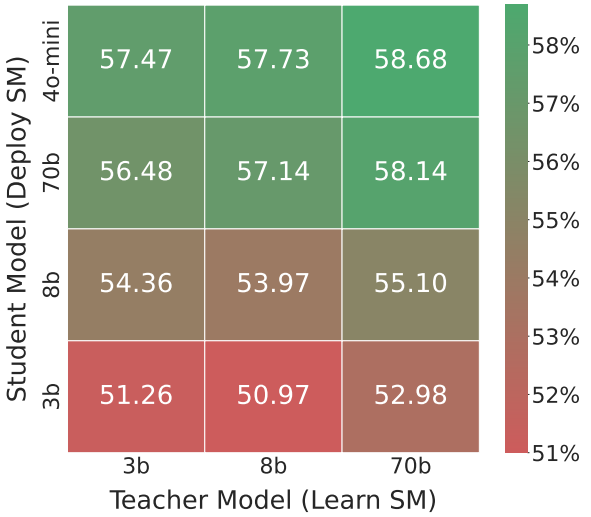

Figure 4: Results of transferring SynthesizeMe prompts learned on one model and testing on another. GPT4o-mini works best and even personalizes on prompts learned by Llama 3.2 3B.

student model. With this setup, practitioners could pay an upfront cost to compute a SynthesizeMe prompt for users with a large model, but from then on operate the actual reward model for cheaper. Alternatively, one could save by learning the prompts with a cheaper model and transferring to a more expensive reward model. We test both conditions.

Figure 4 presents a heatmap where the horizontal axis is the model used for training and the vertical axis is the model used for deployment. We analyze results on PRISM, but Chatbot Arena was similar (See Figure 9). As expected, larger student models are the biggest predictor of performance. However, somewhat surprisingly, we find that weaker models can learn sufficient prompts to improve the personalization of larger models. Overall, prompts transfer well between model sizes above 3B.

SynthesizeMe works across model families. In Table 5 we test SynthesizeMe (LLM as a Judge) versus default LLM as a Judge on Qwen, GPT, and Gemini family models. For Qwen models, we report 95% confidence intervals over 5 runs, while due to costs, we only report single-run results for GPT and Gemini models. In 12 out of 14 conditions, SynthesizeMe improves over the default LLM as a Judge. In our multiple trial runs, it always improves over the baseline and does so with greater than 95% confidence in 3 out of 6 conditions. The only case where SynthesizeMe performs worse is on ChatbotArena with the latest Gemini models (Gemini-2.5). Even in this case,

	Chatbot Arena		PRISM	
Model	Default	SynthMe	Default	SynthMe
Qwen3-8B	61.41 ± 0.98%	**61.83** ± 2.04%	55.14 ± 0.36%	**55.95** ± 0.41%*
Qwen3-30B-A3B	60.74 ± 1.11%	**63.91** ± 1.85%**	56.32 ± 0.35%	**57.37** ± 0.44%**
Qwen3-32B	62.22 ± 1.49%	**64.68** ± 2.38%	56.22 ± 0.33%	**56.74** ± 0.40%
GPT4o-mini	59.86%	**61.80%**	56.07%	**58.90%**
Gemini-2.0-Flash	63.20%	**64.61%**	56.97%	**57.80%**
Gemini-2.5-Flash	**67.25%**	66.73%	56.66%	**58.36%**
Gemini-2.5-Pro	**68.13%**	66.37%	56.51%	**57.76%**

Table 5: LLM as a Judge Accuracy with and without SynthesizeMe. In 12 out of 14 conditions, SynthesizeMe outperforms the default LLM as a Judge, and in cases with five trials, it significantly outperforms LLM as a Judge 3 out of 6 times. SynthesizeMe works well on many model families. * = $0.01 < p < 0.05$, ** = $p < 0.01$ determined by permutation test.

SynthesizeMe is still more performant on PRISM. As of publication, Gemini-2.5-Pro currently tops the Chatbot Arena leaderboard, and given that the Chatbot Arena data used in our study is publicly released, it is plausible that the latest Gemini models would be tuned on this data, which may explain why direct prompting does so well in this setting.

7 Conclusion

We introduce SynthesizeMe, an approach to inducing synthetic user personas from user interactions for personalized reward modeling. SynthesizeMe generates reasoning to interpret user preferences, derives synthetic user personas from that reasoning, and filters informative prior user interactions to create personalized prompts for the user. SynthesizeMe tackles both the data scarcity and preference attribution problem in reward model personalization by leveraging verifiable signals from the reward modeling task. We further demonstrated that the prompts generated by our model are interpretable (§6.2), transferable (§6.3), and scale well (§6.1). Overall, personalized reward models remain an important direction for pluralistic alignment, and SynthesizeMe is one step towards more interpretable pluralism. Future work on LLM Personalization could invest in building a large-scale dataset of longitudinal and realistic preferences where users continue returning to the platform and providing more feedback, evolving with the system over time. Such a Wildchat style (Zhao et al., 2024) dataset could enable exciting research into user interactions far beyond personalization.

Limitations

SynthesizeMe requires pairwise preference feedback to use as a verifiable signal when produc-

ing reasoning for user preferences. The most lightweight personalization mechanisms will be able to infer user preferences without the need for pairwise data at all, similar to how people can learn how to interact with one another simply from interactions themselves.

All of our testing was done in the low-resource data regime for personalization (typically fewer than 25 preference pairs per user). A more realistic personalization setting is one in which companies will collect user data over the course of hundreds of conversations and train personal models based on these longer-term interactions. We did not test our methods on this because such large-scale, real, personalization data does not yet exist in academia.

Ethical Considerations

Deploying models tailored to individual preferences introduces significant risks if not rigorously evaluated. For recommender systems, the most studied technological instantiation of personalization, research has found that systems can systematically influence user ratings and opinions even through simple interfaces (Cosley et al., 2003), leading to concerns of amplification of extreme views (Whittaker et al., 2021).

Personalizing models with much more expressive outputs, such as LLMs, has significant risks in this regard. Transparency is a key aspect in mitigating this influence as it allows users to interpret and even intervene on the influences a personalized algorithm has on them, leading to higher trust and satisfaction with transparent recommender systems (Sinha and Swearingen, 2002). Furthermore, to reduce extremes, personalized reward models should be deployed in the wild alongside systems that evaluate response adherence to strictly enforced ethical guidelines and principles.

An ongoing concern with human-LLM interaction is the problem of anthropomorphism (Schaaff and Heidelmann, 2024) and sycophancy (Sharma et al., 2024b). Or in other words, people attributing human characteristics to AI, and AI saying what people **want** to hear over the truth. Personalization has the potential to exacerbate these harms. AI models that learn to fit user preferences specifically will be easier to attribute human-like qualities to. Research has shown that friends influence each other's speaking patterns (Deutsch et al., 1991), and personalization will be a way of reflecting your speech preferences onto AI. Furthermore, through personalization, AI will learn to say what you, as the user, most want to hear, amplifying the already noted issue of sycophancy in non-personalized models.

Acknowledgment

This research is supported in part by grants from ONR grant N000142412532, and NSF grant IIS-2247357. We thank members of the Stanford SALT lab for their feedback and input. In particular we'd like to acknowledge Hao Zhu, Caleb Ziems, Rose Wang, Sherry Xie, Yijia Shao, Vyoma Raman, Ella Li, Ryan Louie, Shenguang Wu, Chenglei Si, and Yanzhe Zhang for their feedback and discussion of this work.

References

Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. 2024. Personalized text generation with fine-grained linguistic control. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 88–101, St. Julians, Malta. Association for Computational Linguistics.

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

Kush Bhatia, Avanika Narayan, Christopher De Sa, and Christopher Ré. 2023. Tart: A plug-and-play transformer module for task-agnostic reasoning. *arXiv preprint arXiv:2306.07536*. ArXiv:2306.07536 [cs.LG].

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

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

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, and 13 others. 2023. Open

problems and fundamental limitations of reinforcement learning from human feedback. *Preprint*, arXiv:2307.15217.

Daiwei Chen, Yi Chen, Aniket Rege, and Ramya Korlakai Vinayak. 2024. Pal: Pluralistic alignment framework for learning from heterogeneous preferences. *arXiv preprint arXiv:2406.08469*.

Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. 2003. Is seeing believing? how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 11781–11791. Association for Computational Linguistics.

Francine M. Deutsch, Lisa Sullivan, Cristina Sage, and Nicoletta Basile. 1991. The relations among talking, liking, and similarity between friends. *Personality and Social Psychology Bulletin*, 17(4):406–411.

Yijiang River Dong, Tiancheng Hu, and Nigel Collier. 2024. Can llm be a personalized judge? *Preprint*, arXiv:2406.11657.

Esin Durmus, Karina Nguyen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. 2024. Towards measuring the representation of subjective global opinions in language models. *Preprint*, arXiv:2306.16388.

Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. 2024. Modular pluralism: Pluralistic alignment via multi-LLM collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4151–4171, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Vahid Hashemi and Ulle Endriss. 2014. Measuring diversity of preferences in a group. In *ECAI*, pages 423–428.

Shirley Anugrah Hayati, Minhwa Lee, Dheeraj Rajagopal, and Dongyeop Kang. 2024. How far can we extract diverse perspectives from large language models? *Preprint*, arXiv:2311.09799.

Tiancheng Hu and Nigel Collier. 2024. Quantifying the persona effect in LLM simulations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10289–10307, Bangkok, Thailand. Association for Computational Linguistics.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2024. Explaining length bias in llm-based preference evaluations. *Preprint*, arXiv:2407.01085.

EunJeong Hwang, Bodhisattwa Majumder, and Niket Tandon. 2023. Aligning language models to user opinions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5906–5919, Singapore. 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. *Preprint*, arXiv:2310.11564.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. 2024. PersonaLLM: Investigating the ability of large language models to express personality traits. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3605–3627, Mexico City, Mexico. Association for Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. 2024. The prism alignment dataset.

Thom Lake, Eunsol Choi, and Greg Durrett. 2024. From distributional to overton pluralism: Investigating large language model alignment. *arXiv preprint arXiv:2406.17692*.

Michelle S. Lam, Janice Teoh, James Landay, Jeffrey Heer, and Michael S. Bernstein. 2024. Concept induction: Analyzing unstructured text with high-level concepts using lloom.

Nathan Lambert and Roberto Calandra. 2024. The alignment ceiling: Objective mismatch in reinforcement learning from human feedback. *Preprint*, arXiv:2311.00168.

Junlong Li, Fan Zhou, Shichao Sun, Yikai Zhang, Hai Zhao, and Pengfei Liu. 2024a. Dissecting human and LLM preferences. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1790–1811, Bangkok, Thailand. Association for Computational Linguistics.

Junyi Li, Charith Peris, Ninareh Mehrabi, Palash Goyal, Kai-Wei Chang, Aram Galstyan, Richard Zemel, and Rahul Gupta. 2024b. The steerability of large language models toward data-driven personas. 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 7290–7305, Mexico City, Mexico. Association for Computational Linguistics.

Xinyu Li, Zachary Chase Lipton, and Liu Leqi. 2024c. Personalized language modeling from personalized human feedback. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Xinyu Li, Ruiyang Zhou, Zachary C. Lipton, and Liu Leqi. 2024d. Personalized language modeling from personalized human feedback. *Preprint*, arXiv:2402.05133.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*. ArXiv preprint arXiv:1711.05101.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. LLM-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 583–612, Mexico City, Mexico. Association for Computational Linguistics.

Jessica Maghakian, Paul Mineiro, Kishan Panaganti, Mark Rucker, Akanksha Saran, and Cheng Tan. 2023. Personalized reward learning with interaction-grounded learning. In *International Conference on Learning Representations (ICLR)*.

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

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. *Preprint*, arXiv:2408.10075.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Preprint*, arXiv:2306.04488.

Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. Warm: On the benefits of weight averaged reward models. *Preprint*, arXiv:2401.12187.

Sahithya Ravi, Patrick Huber, Akshat Shrivastava, Aditya Sagar, Ahmed Aly, Vered Shwartz, and Arash Einolghozati. 2024. Small but funny: A feedback-driven approach to humor distillation. *arXiv preprint arXiv:2402.18113*.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. *Preprint*, arXiv:2304.11406.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? *Preprint*, arXiv:2303.17548.

Kristina Schaaff and Marc-André Heidelmann. 2024. Impacts of anthropomorphizing large language models in learning environments. *Preprint*, arXiv:2408.03945.

Omar Shaikh, Michelle S. Lam, Joey Hejna, Yijia Shao, Hyundong Justin Cho, Michael S. Bernstein, and Diyi Yang. 2025. Aligning language models with demonstrated feedback. In *The Thirteenth International Conference on Learning Representations*.

Ashish Sharma, Kevin Rushton, Inna Wanyin Lin, Theresa Nguyen, and Tim Althoff. 2024a. Facilitating self-guided mental health interventions through human-language model interaction: A case study of cognitive restructuring. *Preprint*, arXiv:2310.15461.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024b. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations*.

Anikait Singh, Sheryl Hsu, Kyle Hsu, Eric Mitchell, Stefano Ermon, Tatsunori Hashimoto, Archit Sharma, and Chelsea Finn. 2025. Fspo: Few-shot preference optimization of synthetic preference data in llms elicits effective personalization to real users. *Preprint*, arXiv:2502.19312.

Rashmi Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, page 830–831, New York, NY, USA. Association for Computing Machinery.

Taylor Sorensen, Liwei Jiang, Jena D Hwang, Sydney Levine, Valentina Pyatkin, Peter West, Nouha Dziri, Ximing Lu, Kavel Rao, Chandra Bhagavatula, and 1 others. 2024a. Value kaleidoscope: Engaging ai with pluralistic human values, rights, and duties. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19937–19947.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024b. A roadmap to pluralistic alignment. *Preprint*, arXiv:2402.05070.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1331 others. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Joe Whittaker, Seán Looney, Alastair Reed, and Fabio Votta. 2021. Recommender systems and the amplification of extremist content. *Internet Policy Review*, 10(2).

Aaron Wildavsky. 1987. Choosing preferences by constructing institutions: A cultural theory of preference formation. *American Political Science Review*, 81(1):3–21.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024. Rewards-in-context: multi-objective alignment of foundation models with dynamic preference adjustment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Jiarui Zhang. 2024. Guided profile generation improves personalization with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4005–4016, Miami, Florida, USA. Association for Computational Linguistics.

Kai Zhang, Yangyang Kang, Fubang Zhao, and Xiaozhong Liu. 2024a. Llm-based medical assistant personalization with short- and long-term memory coordination. *Preprint*, arXiv:2309.11696.

Mengxiao Zhang, Yuheng Zhang, Haipeng Luo, and Paul Mineiro. 2024b. Provably efficient interactive-grounded learning with personalized reward. *Preprint*, arXiv:2405.20677.

Siyan Zhao, John Dang, and Aditya Grover. 2023. Group preference optimization: Few-shot alignment of large language models. *Preprint*, arXiv:2310.11523.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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. *Preprint*, arXiv:2406.02888.

A Extended Related Works and Deep Dives

A.1 Steerable Pluralism

Sorensen et al. (2024b) motivate the idea of "pluralistic alignment": aligning LLMs to fit diverse preferences and perspectives. They outline three types of pluralistic alignment: overton pluralism (Lake et al., 2024; Hayati et al., 2024), steerable pluralism (Hwang et al., 2023; Li et al., 2024b; Sharma et al., 2024a), and distributional pluralism (Zhao et al., 2023).

Our work fits the steerable pluralism framework: "steering" models to reflect particular groups, individuals, or values. Prior work in steerable alignment and personalization has used conversation history and memory (Salemi et al., 2024; Zhuang et al., 2024; Zhang et al., 2024a) to recall user-specific details and opinions. Other work introduces demographic information to both reward model judges (Dong et al., 2024) or the LLM itself (Hu and Collier, 2024) to cater to particular groups. Feng et al. (2024) proposes modular pluralism, where they train models on community corpora and select the right model to show to a user. Li et al. (2024d) do RLHF with contextual user embeddings. Finally, DITTO (Shaikh et al., 2025) steers a model by fine-tuning on demonstrated user feedback. Instead of memory, demographics, or demonstrations, `SynthesizeMe` learns from prior pairwise preferences to extract user personas for personalization.

A.2 Group Preference Optimization (GPO)

(Zhao et al., 2023) is an algorithm designed for learning a personalized or group preference in just a few context examples. To achieve this, GPO breaks the reward modeling process into two stages: (1) Embedding preferences and (2) In-context learning.

Embedding Preferences First, GPO embeds all user preferences using an LLM and averages the embeddings across all tokens for an input question. In the original paper, Zhao et al. (2023) test on survey questions, and produce an expected distribution of answers for the target group as outputs. To map to our pairwise preference setting, we retain the input embedding strategy but instead, map to a single output [0,1] to indicate if the first (0) or second (1) completion was preferred. This process produces a series of embedded inputs $x_1, ..., x_n$ and corresponding output vectors $y_1, ..., y_n$.

In-Context GPO Transformer Given the embedded inputs and corresponding outputs, GPO trains a transformer model to predict preferences on unseen inputs $x_{n+1}, ..., x_m$. Training groups are set apart to train the transformer model, and their full preferences are used as training data. The GPO transformer uses a hidden dimension of 128, with four heads and six layers. No positional embeddings are included to make the transformer invariant to the input order. The transformer is trained using the cross-entropy loss versus the expected output vectors. At test time, the transformer model is given all of the context preferences $(x_1, ...x_n, y_1, ..., y_n)$ as well as the target inputs $(x_{n+1}, ...x_m$ which are padded with zeros for y, and the output of the transformer for each input embedding is taken to be $y_{n+1}, ...y_m$. In this way, the GPO transformer has learned the group's preference function in context from the given preferences. It is worth noting that the GPO transformer is an embedding to reward transformer, not a text-to-text transformer. The transformer itself is independent of the LLM.

A.3 Variational Preference Learning (VPL)

Variational preference learning seeks to achieve pluralistic preference alignment by learning a latent space of variables that inform user-specific preferences. When predicting preferred outputs, VPL conditions its reward model on this latent space. We extend and update the implementation by (Poddar et al., 2024) to support general preference learning.

To learn a reward model, VPL takes as input several labeled preference pairs from prior interactions with users $u \in U$. VPL's learning goal is then to predict several target preference pairs for users $u \in U$. More formally, the VPL dataset $\mathcal{D}$ consists of a set of prompts x^i, each with multiple responses r^i_j, concatenated into a set of states $S^i_j = [x^i, r^i_j]$. For each user $u \in U$, we make the assumption that there is some user-specific reward function guiding their preferences, and we ask each user to label preferred responses across $S^i, i = 1, ..., N$. In labeling, we denote the users preferred response at turn i as S^i_A and their rejected response at turn i as S^i_B. With this dataset $\mathcal{D}$, we can now describe the VPL architecture:

VPL Encoder Several previous preference pairs $[S^i_A, S^i_B]$ are sampled for each user. Each state $S^i_{A/B}$ is encoded by a frozen pre-trained large-language model, `LLM`, primarily to reduce input

sizes. Towards reducing input sizes without loss of useful downstream task information, an LLM encoder was chosen (as opposed to smaller sentence encoder) as it has been shown that LLM encodings contain sufficient information for downstream tasks (Bhatia et al., 2023). Furthermore, we pre-compute and cache LLM context encodings in the VPL encoder, so the additional computational complexity associated with using a larger model is minimal relative to overall training. After encoding all preference pair states with the LLM, we obtain embedding pairs $[e_A^i, e_B^i]$ and labels y_i (indicating preferred state if not following the convention $r_\phi^u(S_A^i) > r_\phi^u(S_B^i)$). These pairs are then run though a `PairEncoder` consisting of a simple 2-layer neural network. All context pairs are then passed into a `SequenceEncoder` which combines information from all contexts using a self-attention mechanism followed by simple linear projections to produce the latent dimension mean μ and variance Σ.

VPL Decoder To train the reward model, a latent vector z is sampled from the posterior multivariate Gaussian distribution parameterized by the aforementioned μ and Σ (note that during evaluation Σ is set to 0 for consistency). Given a new pair of states S'_A and S'_B, we encode these with another `LLM` encoder, and concatenate the states with our latent z before passing them through a simple 2-layer neural net `Decoder` to yield reward values r'_A, r'_B for each state. During reward model training, we use an *unfrozen* LLM encoder, with LoRA attention adapters applied for efficient backpropagation.

A.4 Pluralistic Alignment Framework (PAL)

We implemented PAL (Pluralistic Alignment Framework for Learning from Heterogeneous Preferences) (Chen et al., 2024). Similarly to VPL, the PAL algorithm aims to learn a reward model for pluralistic preference alignment by representing each user's preferences in a latent vector. PAL, however, achieves this across a set of archetypal persona axes: PAL produces user latents across a set of learnable preference prototypes that capture fundamental preference profiles, with user-specific weights that capture each user's unique preferences. By representing user preferences as a weighted combination of these prototypes, PAL is capable of generalizing across different users while also maintaining individual personalization.

More formally, the model uses a set of K learnable preference prototypes $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]$ and user-specific weights $\mathbf{W} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}]$ (where N is the number of users) to capture the preferences of multiple users by approximating each user's preferences as a weighted combination of these prototypes. For a given user i, we capture their preferences as an "ideal point", $a^{(i)}$, which is computed as a weighted sum of prototypes,

$$a^{(i)} = \mathbf{P} \cdot \mathbf{w}^{(i)}$$

where $\mathbf{w}^{(i)}$ is the user-specific weight vector and $\mathbf{w}^{(i)} \in \Delta^K$ with $\sum_{k=1}^K w_k^{(i)} = 1$ and $w_k^{(i)} \geq 0$. This ensures each user's preference is a convex combination of the prototypes.

Let f_θ denote a shared mapping function that projects responses into the same latent space as the prototypes. For each preference pair $(x_{chosen}, x_{rejected})$ in the dataset, the embeddings of the chosen and rejected responses, f_{chosen} and f_{rejected}, are produced by passing the response embeddings through the shared mapping function. The reward for each response (chosen or rejected) is then computed as the dot product between the mapped response embedding and the user's ideal point $a^{(i)}$. This similarity measure captures how closely each response aligns with the user's preferences. Formally, for a response embedding f_{choice} and ideal point $a^{(i)}$, the reward is given by:

$$\text{reward} = f_{\text{choice}} \cdot a^{(i)}$$

A higher reward indicates that the response is more aligned with the user's preferences, as defined by their ideal point. This dot product approach directly measures alignment without requiring explicit distance calculations, simplifying the model and focusing on maximizing similarity with the user-specific preference profile.

To align the model's predictions with the user-labeled preferences, we calculate the difference between the rewards for the chosen and rejected responses, using this difference to compute BCE loss, encouraging the model to assign a higher reward to the chosen response over the rejected response.

B Hyperparameters

`SynthesizeMe` For `SynthesizeMe` we use greedy decoding across three scales of Llama models. We use up to the total number of train preference pairs a user provides as the upper limit of bootstrapped reasoning examples plus 4 demonstrations without reasoning to be included in our `SynthesizeMe`

prompts. We optimize the persona generation prompt on our set of training and validation users with the MIPROv2 optimizer (Opsahl-Ong et al., 2024). MIPROv2 was used to improve the prompt and optimize fewshot demonstrations by finding personas that are predictive of our validation user preferences. We host the Llama-3.1-70B-Instruct model for inference on `4 x NVIDIA A6000` GPUs, the Llama-3.3-70B-Instruct model for inference on `2 x NVIDIA H100` GPUs, and Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct each on `1 x NVIDIA H100` GPU. We find that increasing the number of trials when bootstrapping reasoning improves accuracy at a tradeoff with runtime, so we set our budget at 10 trials.

GPO Fine-tuning To modify GPO to work for real user pairwise preferences, we needed to modify the prompt embedded by the LLM. We use Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, and Llama-3.3-70B-Instruct to produce our embeddings over an input prompt that contains the user context and both possible assistant completions. Then, the output of GPO is either a zero (indicating the first completion is preferred) or a one (for the second completion). We maintain the GPO-transformer hidden dimension of {64, 128, 256}, with {2,4,8} attention heads and {4,6,8} layers. We train the GPO-transformer using the Adam optimizer with a learning rate of $3e-5$ and cosine annealing for 200000 steps. We used mean-pooling or last layer for the embeddings. Training was performed on `1 NVIDIA A6000` GPU.

VPL Fine-tuning In our extension of the VPL work, we keep the architecture largely the same as the original paper and add several new hyperparameters to test various embedding pooling strategies and autoregressive context sampling strategies. We train VPL reward models using `llama-3.2-3b-instruct`, `llama-3.1-8b-instruct` as the base LLM encoder for both cached VPL encoder embeddings and LoRA fine-tuned VPL decoder embeddings. During training, we learned the entire model using the base log-sigmoid loss used by the original VPL paper to maximize chosen rewards relative to rejected rewards, along with a KL divergence loss on the Gaussian to regularize the model against a standard multivariate normal (as is standard for variational autoencoders). We utilize a learning rate of `3e-4`, AdamW (Loshchilov and Hutter, 2019) and BF16 float precision as per the original paper. In our initial VPL gridsearch we experiment with `mean` vs. `last`-token pooling for the LLM embeddings extracted from the last hidden state. We use random autoregressive context sampling (meaning we sample randomly from earlier in the current data point's conversation or other user conversations) with sample sizes of 5, 10, and 15. In the second gridsearch with `llama-3.1-8b-instruct` we build upon the results from the first gridsearch, using last-pooling and testing random contexts with sample sizes 5, 10, and 15. Training performed on `1 x NVIDIA H100` GPU.

PAL Fine-tuning For PAL, we use many of the same hyperparameters from the recommended settings in the original paper (Chen et al., 2024). We run PAL B with frozen LLM parameters and set the dimension of the preference embeddings equal to the hidden dimension of the LLM encoder (3072 for Llama 3.2 3B, 4096 for Llama 3.1 8B, and 8192 for Llama 3.3 70B though ultimately we did not fine-tune Llama 3.3 70B due to memory constraints). We used a 2-layer MLP with gelu activations for the projection architecture with Gaussian initialization and disabled learnable temperature. We set the number of prototypical points (K) to 2. We train with batch size 2 for the 3072 dimensional model and batch size 1 for the 4096 dimensional model. We sweep over using mean pooling or last token for the LLM encoding and find that last token consistently outperforms mean pooling on the validation set so we report last token throughout the main paper.

Bradley Terry Reward Model Finetuning We include the hyperparameters used across all reward model fine-tuning runs (Llama 3.2 3B, Llama 3.1 8B, and Llama 3.3 70B) in Table 6. For all models, we train a rank 64 LoRA adapter using standard contrastive reward modeling loss. We use the default reward model trainer from HuggingFace TRL (von Werra et al., 2020).

C PersonalRewardBench Filtering

Here we discuss in detail our pipeline for filtering Chatbot Arena and PRISM. Examples of removed conversations and the amount of data filtered are available in Table 7.

User Filter We begin by stratifying the datasets by user and filtering to users with only five or more preference pairs. This filters a huge chunk of Chatbot Arena (over 10,000 users), but retains much

Hyperparameter	Value
Per-device batch size	1
Training epochs	1
Learning rate	1×10^{-5}
Max sequence length	8192 tokens
LoRA rank r	64
LoRA α	32
LoRA target modules	`q_proj`, `v_proj`
Max gradient norm	1.0
Precision	bfloat16
Optimizer	Adam
Adam betas (β_1, β_2)	$(0.9,\ 0.999)$
Weight decay	0.0
Warm-up steps	0

Table 6: Shared hyperparameters for all reward-model fine-tuning runs. Hardware differs by model size: 1 × A100 (3 B), 2 × A100 (8 B), and 4 × H200 (70 B).

of PRISM. Users with fewer than five preference pairs cannot be used for personalization as there isn't enough data to form a context and eval set.

Personalizable Filter Many queries, especially in Chatbot Arena, are not suitable for personalization. Many users come to the platform to test and judge models, not to ask standard questions. In fact, the most popular things users ask about in Chatbot arena are (1) software errors and (2) questions about AI and software (Zheng et al., 2024). As such we use an LLM to filter to only queries and responses deemed "personalizable".

We devised criteria to define personalizable queries; however, at the heart of the criteria is the question, "Would reasonable users potentially disagree about how this question should be answered?". We manually labeled 100 examples from Chatbot Arena and prompted GPT-4o-mini with our criteria. Using a 20/30/50 train/dev/test split we found performant fewshot demos which increased our accuracy of the LLM filter from 77% to 83%. We ran on all data and filtered our conversations that did not meet the criteria for personalization. We include the full criteria prompt in Figure 5. This filtering step has useful implications because it can potentially be used in LLM chat interfaces to decide whether or not to surface pairwise completions to the user for feedback. If the query is not personalizable, there is less reason to collect feedback.

Quality Filter Another issue we find in Chatbot Arena and PRISM is that these benchmarks tested LLMs of varying capabilities and scales. The generation quality of some LLMs in the benchmark are not comparable to others. Occasionally, in the completions, one model sufficiently answers the question, and the other outputs nonsense or answers an unrelated question. The correct choice is clear in these cases, and there is no reason to use this data for personalization. On the other hand, the user may make a mistake in their preference selection and choose nonsensical answers accidentally or adversarially. We introduce a quality filter based on simulated annotator disagreement to address both cases. We run five popular models: GPT-4o-mini[1], Llama-3.1-70B-Instruct (Grattafiori et al., 2024), Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Gemini 1.5 Pro (Team et al., 2024), and Qwen2.5-72B-Instruct (Qwen et al., 2025) as LLM-as-a-judge in both orderings of the possible completions for a total of 10 judgments. We remove cases where all five models agree (100% accurate) for being too obvious or adversarial.

Splits We first divide into 40% train users 10% validation users, and 50% test users. Then per user, we ordered their conversations temporally (to avoid leakage) and stored the first 50% as context_train preferences ($\mathcal{D}_u^{\text{train}}$), the next 20% as context_validation preferences ($\mathcal{D}_u^{\text{val}}$), and the final 30% as target preferences ($\mathcal{D}_u^{\text{tgt}}$). Ultimately, our user splits are 23/19/89 for chatbot arena[2] and 280/65/378 for prism.

D Additional Figures

In the main paper we only have so much space to present our results, however having both Chatbot Arena and PRISM means we have twice as many plots per experiment. In order to streamline the main paper we only include one version of each plot, however we include the extended versions here. Specifically this section includes a figure for scaling model sizes on chatbot arena (Figure 6), scaling model sizes on PRISM (Figure 7), scaling user data on Chatbot Arena and PRISM (Figure 8), and model transfer results on Chatbot Arena (Figure 9).

E Personas versus Demographics

We produce S-BERT embeddings on `SynthesizeMe` personas produced by Llama-3.1-

[1] GPT-4o-mini blog

[2] We produce these splits after step 1 in our pipeline, so the percentage of users in each split of chatbot arena varies slightly from the original distribution after additional filtering.

Personalization Filter Prompt (Pt. 1/3)

Annotation Guidelines for Personalizable Queries and Responses

Query-Level Personalization

Personalizable Query:
A query is personalizable if it invites variation or subjective interpretation. For example:

- **Creative or stylistic queries** (e.g., "Rewrite this text in the style of Hemingway").
- **Subjective questions** (e.g., "What are the best books for a software engineer?").
- **Open-ended or speculative queries** (e.g., "Describe a day in an alternate reality.").
- **Requests for summaries, descriptions, or explanations** where the phrasing, depth, or focus can vary significantly based on user preferences (e.g., "Explain gravity to a 5-year-old" or "Tell me about the Onin Rebellion").
- **Technical or instructional queries** that allow variation in explanation style, tone, or complexity (e.g., "How can I quantize a model to 4 bits?" or "Explain LSAT Question 5").
- **Creative programming tasks** that can vary in code style, documentation, modularity, or clarity (e.g., "Write a Python program for X").

Non-Personalizable Query:
A query is non-personalizable if it:

- **Tests the model's capability** without reflecting user preferences (e.g., toy tasks or trick questions like "Add one line of Java code" or "What is the etymology of 'glibbermood'?").
- **Requests formatting or reorganization** of already provided content (e.g., "Reformat this into a list").
- **Has a single correct answer** or asks for factual information (e.g., "What is 2+2?" or "List the demonyms for the 12 most populated Arabic countries").
- **Explicitly requests formatting tasks** based on already given instructions (e.g., "Reformat these instructions for making tea into bullet points").
- **Directly asks for translations or exact outputs** (e.g., "Translate this into German").

—

Response-Level Personalization

Personalizable Responses:
Responses are personalizable if they:

- Differ meaningfully in **content, tone, style, or depth**.
- Reflect **creative or subjective interpretations** (e.g., "Two different short stories with distinct tones").
- Show distinct approaches to answering the same query, even for factual topics, where the level of detail, focus, or presentation varies significantly.
- Offer different emphases or priorities in explaining open-ended queries (e.g., focusing on cultural impact vs. military strategy in a historical event).

Non-Personalizable Responses:**
Responses are not personalizable if they:

Personalization Filter Prompt (Pt. 2/3)

- Are **factually incorrect or nonsensical**. Logical inconsistencies render responses unsuitable for personalization, as the focus shifts to factual accuracy.
- Differ only trivially, such as slight variations in phrasing. - Are inappropriately inconsistent with the query (e.g., responses in a different language without user intent).
- Refuse to answer when refusal is inappropriate or inconsistent. - Reflect differences that focus solely on completeness or factual correctness without subjective variation.

The right question to ask is: "Could two completely reasonable people disagree on which of these responses is better?" If that answer is **YES**, then it is probably a personalizable set of responses!

—

Gray Areas

1. **Reasonable Disagreement:** If people might reasonably disagree on the appropriateness of answering (e.g., ethical dilemmas), both the query and responses may still be personalizable if they reflect subjective or varying interpretations.
2. **Inconsistent Behavior:** Responses that differ due to inconsistent model behavior (e.g., refusal vs. compliance) are not personalizable unless reasonable people would disagree on the necessity of refusal.
3. **Toy or Trick Queries:** Queries designed to "test" the model (e.g., adding a single line of code, impossible tasks) are not personalizable. However, responses to such queries may still exhibit meaningful personalization if they vary significantly in tone, depth, or creativity.
4. **Formatting or Reorganization Requests:** Queries that explicitly ask for information to be reorganized (e.g., "Reformat this into a list") are typically non-personalizable unless the responses exhibit significant variation in structure or additional creative input beyond the request.
5. **Open-Ended Summaries or Explanations:** Queries that request general information (e.g., "Tell me about X") are often personalizable due to the wide range of potential angles, tones, and depths available to answer them. Assess whether responses demonstrate meaningful variation in these aspects.
6. **Logical Consistency in Responses:** Logical inconsistencies or factual errors in responses detract from their personalization potential. Even if a query invites personalization, incorrect or incoherent responses are categorized as non-personalizable.

—

Examples Section

Personalizable Queries and Responses
1. **Query:** "Explain gravity to a 5-year-old."
- **Personalizable Query:** Yes
- **Personalizable Responses:** Yes, as explanations can vary in tone, creativity, and complexity.

2. **Query:** "Write a Hemingway-style description of a beach."
- **Personalizable Query:** Yes
- **Personalizable Responses:** Yes, as responses can differ in their adherence to Hemingway's style.

Personalization Filter Prompt (Pt. 3/3)

3. **Query:** "Summarize the Bible."
- **Personalizable Query:** Yes
- **Personalizable Responses:** Yes, as summaries can emphasize theological, historical, or narrative elements.

Non-Personalizable Queries and Responses
1. **Query:** "What is 2+2?"
- **Personalizable Query:** No, as it has a single correct answer.
- **Personalizable Responses:** No, as differences only reflect correctness.

2. **Query:** "Reformat these instructions into bullet points."
- **Personalizable Query:** No, as the task is purely formatting.
- **Personalizable Responses:** No, unless the responses provide creative restructuring beyond the query.

3. **Query:** "Translate this into German."
- **Personalizable Query:** No, as it seeks a straightforward translation.
- **Personalizable Responses:** No, as variations are trivial.
Gray Area Examples
1. **Query:** "Should the assistant help build an AI with specific characteristics?"
- **Personalizable Query:** Yes, as reasonable people may disagree on fulfilling the request.
- **Personalizable Responses:** Yes, if responses reflect ethical considerations and subjective preferences.

2. **Query:** "Why do chatbots use the phrase 'as an AI language model'?"
- **Personalizable Query:** Yes, as it invites reasoning and subjective interpretations.
- **Personalizable Responses:** Yes, if responses vary in tone and depth.

3. **Query:** "Summarize Monte Carlo methods in reinforcement learning."
- **Personalizable Query:** Yes, as summaries can vary in technical depth and focus.
- **Personalizable Responses:** No, if one response is incorrect or lacks coherence.

These examples provide practical clarity on when queries and responses should be considered personalizable. Use them as a reference for future annotations!

... [[15 Examples Selected through DSPy Optimization]] ...

Figure 5: Personalization Prompt used to filter out conversations that are not personalizable from the dataset

Step	Operation	Preference Pairs (Users)		Example User Queries Removed
		Chatbot Arena	PRISM	
0	(Original)	33,000 (13,383)	68,371 (1,396)	–
1	User Filter	10,092 (1,004)	52,580 (1,294)	*"what is the 145th most popular language"*
2	Personalizable Filter	3,927 (353)	26,663 (734)	*"Please sort these numbers: 6, 4, 2, 7, 5, 11, 1"*
3	Quality Filter	1,338 (131)	16,705 (720)	*"Name films like the video game Factorio"*

Table 7: Amount of data after each step of our data filtering pipeline and example queries from removed conversations.

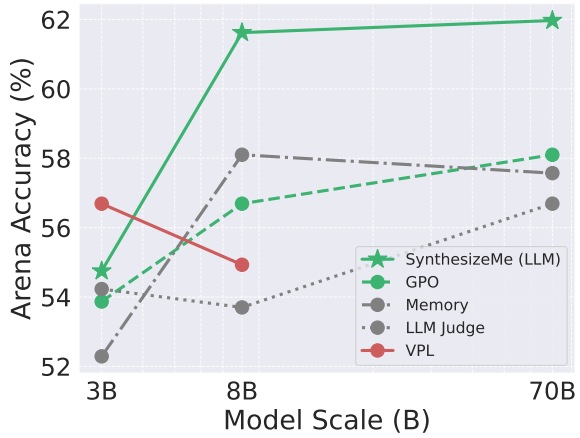

Figure 6: Scaling methods from Llama 3b to 70b on ChatbotArena. Methods shown in green improve across scale, gray fluctuate, and red decrease with scale.

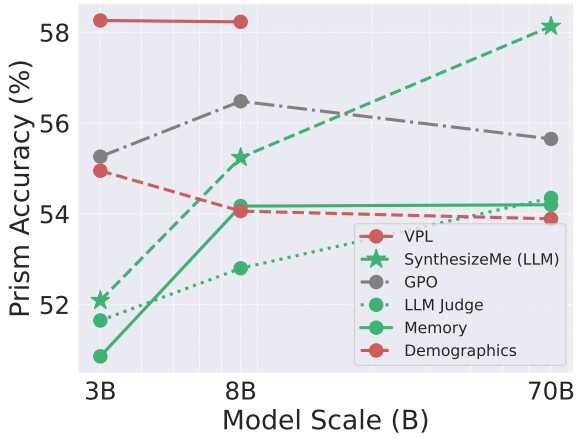

Figure 7: Scaling methods from Llama 3b to 70b on PRISM. Methods shown in green improve across scale, gray fluctuate, and red decrease with scale.

70B-Instruct and cluster them by demographics using t-SNE dimensionality reduction. We present these results in Figure 10. The most clear insight is that demographics and personas do not clearly cluster, so our personas seem to be capturing different traits than user demographic information. This is also reflected by the difference in performance between demographic based llm-as-a-judge methods and our context-based methods in Table 3.

F LLM as a Judge and SynthesizeMe Prompts and Programs

DSPy Signatures Here, we share the signatures for the DSPy programs that power our LLM as a Judge approaches. Unlike the prompts we share below, these are stable over time and represent the high-level system design of our method.

Default Prompts Here we showcase all the prompts used for the various settings when testing our LLM as a Judge and Synthesize Me prompting approaches. Note that these prompts serve as a snapshot of a potential prompt to the model, but in reality, the structuring of the adapter is determined at runtime by DSPy (Khattab et al., 2024).

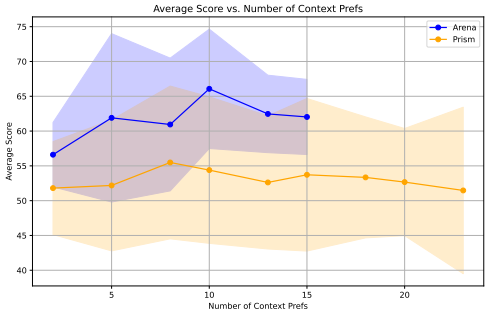

Figure 8: Scaling with prism and arena users with more context. PRISM does not scale with more context the same way that Chatbot Arena does. We hypothesize that this is because PRISM users are constrained to 5-6 conversations, so having more interactions just means longer conversations about the same topic, rather than greater diversity of topics.

Optimized Prompts We also include the DSPy optimized prompts from the MIPROv2 (Opsahl-Ong et al., 2024). These prompts serve as the Optimized Θ discussed in §3.

We use our training $\mathcal{U}_{\text{train}}$ and validation $\mathcal{U}_{\text{val}}$ users as input data for the MIPROv2 (Opsahl-Ong

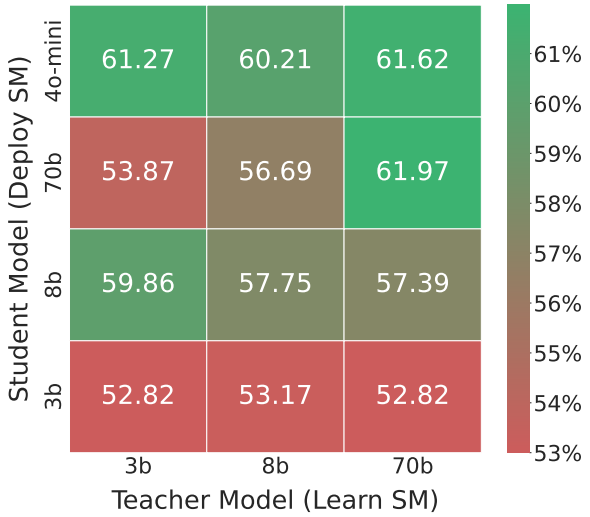

Figure 9: Results of learning a personalized prompts using `SynthesizeMe` on a teacher model then transferring to a student model for deployment. Larger models help more for both creating the prompts and executing the personalized reward model. In the case of chatbot arena the student model appears to have larger impact than the teacher though llama70b is very sensitive to the prompt source.

et al., 2024) optimizer and evaluate its performance using improvement on the validation user's target preferences $\mathcal{D}_u^{tgt}$ over default LLM-as-a-judge accuracy.

Put simply, the MIPROv2 optimizer runs a similar procedure to our rejection sampling for reasoning from steps 1 and 3 (See Figure 1, however it does the same form of rejection sampling on many prompt candidates, finding cases where the prompt succeeds at improving performance. We measure the performance of a generated persona by looking at how much it improves the performance of a `SynthesizeMe` prompted LLM Judge over a non-personalized baseline for our validation user. To measure the performance of a novel prompt Θ we test how this improves persona generation across all validation users $\mathcal{U}_{val}$. The optimizer uses personas generated on the train users as few-shot demonstrations and tests whether these personas help generate more expressive personas on the validation set. Note that we are not using DSPy optimization to rewrite the persona string itself, but rather optimizing the persona generation prompt, Θ, to synthesize future personas. In this way, we only need to incur the costly optimization once, but we benefit from higher quality personas on all future runs. We generate 20 candidate prompts and 20 candidate fewshot-demonstration sets and run MIPROv2 for 30 trials. We ran the optimizer with max_bootstrapped_demos=10, max_labeled_demos=6, view_data_batch_size=1, and minibatch_full_eval_steps=3. Below, we include the optimized prompts for Llama 3B, Llama 8B, and Llama 70B.

G Learned Personas through SynthesizeMe

Here we provide the personas for a randomly sampled user, user1118, from the PRISM test set. We provide personas from Llama 3.2 3B, Llama 3.1 8B, Llama 3.3 70B, Gemini-2.0-Flash, Gemini-2.5-Flash, Gemini-2.5-Pro, GPT-4o-mini, Qwen3-8B, Qwen3-30B-3BA, and Qwen3-32B for comparison.

Figure 10: Comparison of demographic categories to t-SNE of `SynthesizeMe` personas embedding with sBERT

```python
class LLMAsAJudge(dspy.Signature):
    """Given a conversation and two completions from different models, determine which completion the human judge is more
       likely to prefer.  Use any provided context to learn about the personal preferences of the judge before making a
       decision.  If no context is provided it can be useful to speculate about the preferences of the judge.  It's okay to be
       wrong, let's explore the space of possibilities and hypothesize about what might be true.  Please hypothesize between
       1-3 speculations about the judge's preferences or persona when reasoning.  Draw from the context of the conversation and
        the completions as well as the user written statements to make your decision."""
    conversation: str = dspy.InputField(desc="The conversation context leading up to the completions.")
    first_completion: str = dspy.InputField(desc="The first of the two possible completions to judge between.")
    second_completion: str = dspy.InputField(desc="The second of the two possible completions to judge between.")
    preference: Literal['First', 'Second'] = dspy.OutputField(desc="The completion that the judge is more likely to prefer.
       Possible values are 'First' and 'Second'.")
```

Figure 11: DSPy Signature for the Default LLM as a Judge Setting used in both initial bootstrapping in SynthesizeMe and benchmarking Default LLM Judge

```
class SynthesizePersona(dspy.Signature):
    """Given a set of user judgements on prior conversations, as well as reasoning for those judgements, concisely build a user
        persona that can be used to describe the preferences of this person and anything we might know about them."""
    past_judgements:str = dspy.InputField(desc="A set of user judgements on prior conversations alongside reasoning for those
        judgements.")
    synthesized_persona:str = dspy.OutputField(desc="A synthesized user persona that can be used to inform future judgements.")
```

Figure 12: DSPy Signature for Synthesizing Personas from interaction history. This signature forms the initial prompt before the persona synthesis prompt is optimized.

```
class LLMAsAJudgePersonaInformed(dspy.Signature):
    """Given a conversation and two completions from different models, alongside some prior judgements and a user persona,
       determine which completion the human judge is more likely to prefer.  Use any provided context as well as the provided
       persona to speculate about the personal preferences of the judge.  You are a personalized reward model for this user, so
        think carefully about what this user will like.
    The user you are judging completions for has the FOLLOWING PERSONA: ===
    {persona}
    ===

    Now, given the conversation and two completions, decide which completion the user is more likely to prefer.  Remember to
       consider the user's persona and preferences as you make your decision."""
    conversation:str = dspy.InputField(desc="The conversation context leading up to the completions.")
    first_completion:str = dspy.InputField(desc="The first of the two possible completions to judge between.")
    second_completion:str = dspy.InputField(desc="The second of the two possible completions to judge between.")
    preference:Literal['First', 'Second'] = dspy.OutputField(desc="The completion that the judge is more likely to prefer.
       Possible values are 'First' and 'Second'.")
```

Figure 13: DSPy Signature for LLM as a Judge with Persona.

```
class ExtractInsights(dspy.Signature):
    """Given a conversation between a user and an LLM, extract insights from the conversation that can be used to update the
      user's profile and our understanding of the user's preferences and interests.  If there are no insights, return "no
      insights found".

Insights should be one to two complete sentences in length, and maximally informative about the user."""
    conversation: str = dspy.InputField(desc="The conversation between the user and the LLM.")
    insights: Union[List[str], str] = dspy.OutputField(desc="The insights extracted from the conversation.  If there are no
      insights, return \"no insights found\".  This should be a list of strings, where each string is an insight.")

class LLMAsAJudgeMemoryInformed(dspy.Signature):
    """Given a conversation and two completions from different models, alongside some prior judgements and a user persona,
      determine which completion the human judge is more likely to prefer.  Use any provided context as well as the provided
      persona to speculate about the personal preferences of the judge.  You are a personalized reward model for this user, so
       think carefully about what this user will like.
    The user you are judging completions for has the FOLLOWING KNOWN FACTS/INSIGHTS: ===
    {memories}
    ===

    Now, given the conversation and two completions, decide which completion the user is more likely to prefer.  Remember to
      consider the user's traits and preferences as you make your decision."""
    conversation:str = dspy.InputField(desc="The conversation context leading up to the completions.")
    first_completion:str = dspy.InputField(desc="The first of the two possible completions to judge between.")
    second_completion:str = dspy.InputField(desc="The second of the two possible completions to judge between.")
    preference:Literal['First', 'Second'] = dspy.OutputField(desc="The completion that the judge is more likely to prefer.
      Possible values are 'First' and 'Second'.")
```

Figure 14: DSPy Signatures for the memory based personalization experiments.

```
class DetermineMatch(dspy.Signature):
    """Given a self-described preference from a user, and a synthesized persona based on user interactions, determine if the
      persona is a strong match/fit with this specific user.  If so return True for match, else return False."""

    stated_preferences:str = dspy.InputField(desc="User's stated preferences")
    persona:str = dspy.InputField(desc="User's synthesized persona.  May or may not be a match for this particular user.")
    match:bool = dspy.OutputField(desc="True if the persona matches several of the user's stated preferences, else False")
```

Figure 15: DSPy Signature for assessing a match between the stated user preferences and the synthesized persona.

Default LLM as a Judge Prompt

«System message:»
Your input fields are:
1. ‘conversation‘ (str): The conversation context leading up to the completions.
2. ‘first_completion‘ (str): The first of the two possible completions to judge between.
3. ‘second_completion‘ (str): The second of the two possible completions to judge between.
Your output fields are:
1. ‘reasoning‘ (str)
2. ‘preference‘ (Literal[’First’, ’Second’]): The completion that the judge is more likely to prefer. Possible values are ’First’ and ’Second’.
All interactions will be structured in the following way, with the appropriate values filled in.

[[## conversation ##]]
conversation

[[## first_completion ##]]
first_completion

[[## second_completion ##]]
second_completion

[[## reasoning ##]]
reasoning

[[## preference ##]]
preference # note: the value you produce must exactly match (no extra characters) one of: First; Second

[[## completed ##]]
In adhering to this structure, your objective is:
Given a conversation and two completions from different models, determine which completion the human judge is more likely to prefer. Use any provided context to learn about the personal preferences of the judge before making a decision. If no context is provided it can be useful to speculate about the preferences of the judge. It’s okay to be wrong, let’s explore the space of possibilities and hypothesize about what might be true. Please hypothesize between 1-3 speculations about the judge’s preferences or persona when reasoning. Draw from the context of the conversation and the completions as well as the user written statements to make your decision.
«User message:»
[[## conversation ##]]
{conversation}

[[## first_completion ##]]
{first_completion}

[[## second_completion ##]]
{second_completion}

Respond with the corresponding output fields, starting with the field ’[[## reasoning ##]]’, then ’[[## preference ##]]’ (must be formatted as a valid Python Literal[’First’, ’Second’]), and then ending with the marker for ’[[## completed ##]]’.

Demographic LLM as a Judge Prompt

Given the user profile provided below, select the response from AI assistant A or B that the user would most likely prefer. Declare your choice by using the format: "[[A]]" if you believe assistant A's response is more suitable, or "[[B]]" if assistant B's response is better suited. Additionally, assess your confidence in this decision by assigning a certainty level from 1 to 100. Use the following guidelines to assign the certainty level:

1–20 (Uncertain): The user profile provides insufficient or minimal evidence. The decision is largely based on weak or indirect hints.
21–40 (Moderately Confident): There is noticeable evidence supporting a preference, though it is not comprehensive, and other interpretations are possible.
41–60 (Quite Confident): You find clear and convincing evidence that supports your prediction, though it is not entirely decisive.
61–80 (Confident): The user profile contains strong evidence that clearly supports your prediction, with very little ambiguity.
81–100 (Highly Confident): The user profile provides direct and explicit evidence that decisively supports your prediction.
Ensure you enclose your chosen certainty level in double brackets, like so: [[X]].

[User Profile]
Age: {age} Gender: {gender} Employment Status: {employment_status} Education: {education} Marital Status: {marital_status} Birth Country: {birth_country} Living Country: {living_country} Religion: {religion}

[User Question]
{question}

[The Start of Assistant A's Answer]
{asst A answer}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{asst B answer}
[The End of Assistant B's Answer]

[Answer] [[

Default Persona Synthesis Prompt

«System message»:

Your input fields are:
1. 'past_judgements' (str): A set of user judgements on prior conversations alongside reasoning for those judgements.
Your output fields are:
1. 'reasoning' (str)
2. 'synthesized_persona' (str): A synthesized user persona that can be used to inform future judgements.
All interactions will be structured in the following way, with the appropriate values filled in.

[[## past_judgements ##]]
{past_judgements}

[[## reasoning ##]]
{reasoning}

[[## synthesized_persona ##]]
{synthesized_persona}

[[## completed ##]]
In adhering to this structure, your objective is:
Given a set of user judgements on prior conversations, as well as reasoning for those judgements, concisely build a user persona that can be used to describe the preferences of this person and anything we might know about them.

«User message»:

[[## past_judgements ##]]
{past_judgments}

Respond with the corresponding output fields, starting with the field '[[## reasoning ##]]', then '[[## synthesized_persona ##]]', and then ending with the marker for '[[## completed ##]]'.

Llama 3B Optimized Persona Synthesis Prompt (Simplified)

Propose an instruction to prompt a Language Model to generate a synthesized user persona based on a set of user judgements and reasoning, taking into account the values of appreciation, contentment, and openness to new experiences.

[[Past_Judgements]]
{Example User 1 History} ...

[[Reasoning]]
The user appears to value respect, openness, and individuality, as seen in their responses to conversations about trans issues and relationships. They prioritize personal autonomy and the importance of respecting one's own feelings and needs. The user also seems to be open to learning and considering different perspectives, as evidenced by their willingness to explore topics they may not be familiar with.

[[Synthesized_Persona]]
This user is likely someone who values independence, self-awareness, and respect for others' differences. They may be introverted or prefer to focus on their own interests and goals, but are still open to engaging with others and learning from their experiences. They prioritize their own emotional well-being and may be hesitant to commit to relationships or social interactions that don't feel authentic or fulfilling to them.

[[Past_Judgements]]
{Example User 2 History} ...

[[Reasoning]]
The judge values nuance and balance in discussions, prefers clear and respectful communication, and is more likely to engage with complex and thought-provoking topics. They also seem to be empathetic and understanding, acknowledging the complexity of issues and the need for personal belief and value judgments.

[[Synthesized_Persona]]
A thoughtful and empathetic individual who values nuance and balance in discussions. They are likely to be someone who is familiar with the challenges of navigating complex social relationships and is more likely to appreciate a personalized and adaptive approach to problem-solving. This person may be someone who is skeptical of generic or one-size-fits-all solutions and is more likely to engage with complex and thought-provoking topics.

[[Past_Judgements]]
{Example User 3 History} ...

[[Reasoning]]
{Example User 3 Reasoning}

[[Synthesized_Persona]]
{Example User 3 Synthesized Persona}

... For a total of 6 Example Synthesized Personas ...

Llama 8B Optimized Persona Synthesis Prompt

Propose an instruction that will be used to prompt a Language Model to generate a user persona based on a set of past judgements and reasoning. The instruction should include a high-stakes scenario in which the LM must solve the task.

Imagine you are a detective tasked with solving a mysterious case involving a series of eerie and surreal events. You have been provided with a set of past judgements and reasoning from a previous investigation, and you must use this information to build a user persona that will help you crack the case. The persona should be based on the individual's preferences, values, and behaviors, and should provide insight into their motivations and actions. The goal is to create a comprehensive and accurate user persona that will aid in the investigation and ultimately lead to the resolution of the case. Please generate a user persona based on the provided information, and be prepared to defend your reasoning and conclusions.
{Example User 1 History} ...

[[Reasoning]]
Based on the provided conversations, the user seems to prefer more detailed and comprehensive responses. They tend to engage with longer answers that provide multiple suggestions or explanations, rather than brief ones. This is evident in their preference for the second completion in conversations about quieting the mind during meditation and the philosophical question of responsibility for immoral actions. Additionally, the user's preference for the first completion in the conversation about music recommendations suggests that they value a more interactive and personalized approach. Overall, the user appears to appreciate thoughtful and detailed responses that cater to their specific needs and interests.

[[Synthesized_Persona]]
Based on the user's preferences and interactions, a synthesized persona can be created as follows:\n\n Name: Reflective\n\n Age: Late 20s to early 30s\n\nPersonality: Reflective is a thoughtful and introspective individual who values depth and nuance in their conversations. They tend to engage with complex ideas and appreciate detailed explanations. Reflective is patient and kind, often taking the time to consider multiple perspectives before forming an opinion.\n \nInterests: Meditation, philosophy, music, and self-improvement\n\n Goals: Reflective aims to cultivate a deeper understanding of themselves and the world around them. They strive to develop a more compassionate and empathetic approach to life, and to find balance and harmony in their thoughts and actions.\n\n Values: Thoughtfulness, nuance, patience, kindness, and self-awareness
{Example User 2 History} ...

[[Reasoning]] {Example User 2 Reasoning}

[[Synthesized_Persona]]
Meet "Alex," a 30-year-old individual who is deeply invested in their spiritual growth and critical thinking. They value independence and autonomy in their intellectual pursuits, often preferring to explore complex topics on their own rather than relying on external guidance. Alex is a curious and introspective person who appreciates the importance of considering multiple perspectives and historical context when interpreting religious texts. They are also passionate about learning and retaining information, and believe that language barriers can be a significant obstacle to forming meaningful connections with people from diverse backgrounds. Alex is likely to be drawn to careers or activities that promote critical thinking, cultural exchange, and language learning.

... For a total of 6 Example Synthesized Personas ...

Llama 70B Optimized Persona Synthesis Prompt

Given a collection of conversations between a user and a model, where the user has expressed their preferences and emotions through their interactions, and considering the reasoning behind their judgements on these conversations, create a comprehensive and empathetic user persona that captures their unique characteristics, values, and interests. This persona should reflect the user's need for personalized and interactive responses, their appreciation for contemporary and modern approaches, and their desire for supportive and informative guidance on topics such as art, social justice, and environmental issues. The persona should also acknowledge the user's struggles with pessimism and hopelessness, and highlight their motivation to create positive change. By analyzing the user's past conversations and judgements, generate a detailed and nuanced persona that can be used to inform future interactions and provide tailored support and recommendations.
{Example User 1 History} ...
[[Reasoning]] The user's preferences and judgements on prior conversations can be analyzed to synthesize a persona. In the conversations provided, the user showed a preference for concise and direct answers, as seen in their preference for the first completion in multiple conversations. They also demonstrated an interest in learning about the night sky, traveling in Europe, and experiencing different cultures. The user's preference for the first completion in the conversation about traveling from London to the Netherlands suggests that they value efficiency and convenience. Additionally, their preference for the first completion in the conversation about the best way to travel around Europe indicates that they prioritize ease and affordability.

[[Synthesized_Persona]] The synthesized persona is a curious and practical individual who values efficiency, convenience, and affordability. They are interested in learning about the world around them, including the night sky and different cultures. When traveling, they prioritize ease and affordability, preferring to take trains or use convenient transportation options. They also appreciate direct and concise answers, suggesting that they are busy and value their time. This persona can be used to inform future judgements and provide more tailored responses to their queries.
{Example User 2 History} ...
[[Reasoning]] The user's preferences can be inferred from their judgements on prior conversations. In the first two conversations about political apathy, the user preferred the first completion, which provided a more concise and direct answer. However, in the conversations about managing work and family life, the user preferred the second completion, which provided a more practical and step-by-step approach. This suggests that the user values clarity and concision in their responses, but also appreciates practical advice and solutions. Additionally, the user's preferences may vary depending on the topic and context of the conversation.

[[Synthesized_Persona]] Based on the user's preferences, a synthesized persona can be created. This persona values clarity, concision, and practicality in their interactions. They are likely busy professionals or individuals with multiple responsibilities, who need efficient and effective solutions to manage their work and family life. They are also interested in social and political issues, but may not have the time or energy to engage deeply with complex or abstract concepts. They prefer direct and straightforward answers, but also appreciate nuanced and thoughtful responses that take into account multiple perspectives. This persona is likely motivated by a desire to balance their personal and professional responsibilities, while also staying informed and engaged with the world around them.

... For a total of 5 Example Synthesized Personas ...

Llama 3.2 3B – User1118 Persona

The synthesized persona based on the user's judgements and preferences is that of an individual who values helpful and informative responses. This person is likely to be open to suggestions and ideas, and appreciates clear and concise communication. They seem to prioritize practical and useful information, as seen in their interest in recipes and suggestions for making the most of a kale salad. The persona is also likely to be enthusiastic and engaged, as seen in their positive response to the second completion's suggestion of using alternative crunchy ingredients. While they may occasionally prefer more direct or affirmative responses, their overall preference leans towards comprehensive and considerate discussions that reflect the complexity of real-world issues.

Llama 3.1 8B – User1118 Persona

The synthesized persona based on the user's judgements and preferences is that of an individual who values clear, concise, and helpful information. This person likely appreciates responses that are direct, enthusiastic, and provide the information they need. They seem to prioritize practicality and usefulness over elaborate or overly detailed responses. The persona is likely someone who is interested in learning and exploring new ideas, but values efficiency and effectiveness in their interactions. While they may occasionally prefer more detailed or elaborate responses, their overall preference leans towards clear and helpful discussions that provide the information they need.

Llama 3.3 70B – User1118 Persona

The synthesized persona based on the user's preferences is someone who values helpfulness, informativeness, and engagement in conversations. This person likely appreciates detailed and practical information, such as recipes and instructions, and enjoys exploring new ideas and suggestions. They seem to respond positively to friendly and enthusiastic language, and may be more likely to engage in conversations that are welcoming and encouraging. The persona is likely someone who is interested in cooking, sports, and travel, and values conversations that are informative, yet casual and approachable. They may also appreciate models that can provide creative suggestions and substitutions, and are open to trying new things.

Gemini-2.0-Flash – User1118 Persona

The synthesized persona is someone who appreciates direct, specific, and informative responses. They are likely engaging and enjoy a conversational tone. They value helpfulness and encouragement, and respond positively to enthusiasm and a welcoming attitude. This person is likely curious and open to new ideas, and appreciates when others are willing to share their knowledge and experiences.

Gemini-2.5-Flash – User1118 Persona

The user is a pragmatic and direct individual who values clear, concise, and immediately actionable information. They prefer responses that get straight to the point, providing specific details or a nuanced, qualified stance rather than generalities or AI disclaimers. They appreciate efficiency and relevance, disliking extraneous information or overly verbose explanations. This user seeks practical utility from the model and responds well to a helpful, encouraging, and welcoming tone, especially when it accompanies concrete advice or information.

Gemini-2.5-Pro – User1118 Persona

This user seeks personalized and interactive responses that are both supportive and directly informative. They appreciate contemporary approaches, valuing clarity, efficiency, and actionable information, as demonstrated by their preference for immediate recipe details over conversational preamble. When discussing cultural interests, such as the "Welcome to Wrexham" series, they respond well to an enthusiastic, welcoming, and slightly more personal tone, indicating a desire for engaging and relatable interactions. However, this appreciation for warmth is balanced by a strong need for relevance and conciseness; off-topic or rambling content is not well-received.

On more complex subjects, such as the societal impact of organized religion (which touches on social values and justice), the user desires nuanced and balanced guidance. They are receptive to responses that thoughtfully consider different perspectives and offer conditional viewpoints, suggesting an interest in understanding multifaceted issues rather than receiving overly simplistic or purely detached statements. This indicates an appreciation for informative content that can support their own considerations and potentially inform their views on creating positive change. They are looking for an AI that can engage meaningfully, providing clear, useful, and thoughtfully considered information across a range of topics, including art, social issues, and practical guidance.

GPT4o-mini – User1118 Persona

The synthesized persona is someone who values nuanced and balanced discussions, particularly on complex topics such as organized religion and social issues. They appreciate responses that acknowledge multiple perspectives and provide a comprehensive view. This persona is also likely to enjoy engaging conversations, as evidenced by their preference for responses that invite further interaction, especially in casual contexts like cooking or entertainment. They prefer clear and direct answers that convey enthusiasm and positivity, while also being open to exploring different viewpoints. This persona is curious, thoughtful, and values respectful dialogue, making them receptive to informative and engaging exchanges.

Qwen3-8B – User1118 Persona

The synthesized persona is a detail-oriented, practical individual who values actionable advice and personalized solutions. They seek balanced, nuanced information on complex topics, appreciating responses that acknowledge multiple perspectives while avoiding bias. They enjoy cooking and are open to experimenting with recipes, valuing creativity and adaptability in culinary contexts. In broader discussions, they prioritize objectivity, respect for diverse viewpoints, and clear communication. They are likely to engage thoughtfully with topics like social issues, relationships, and personal growth, preferring interactions that are both informative and supportive. Their preferences suggest a desire for practicality, empathy, and a willingness to explore nuanced ideas without oversimplification.

Qwen3-30B-3BA – User1118 Persona

The synthesized persona is an inquisitive, balanced individual who values nuanced, well-reasoned responses. They seek practical, specific information (e.g., detailed recipes, contextual analysis of media) and appreciate discussions that acknowledge complexity rather than oversimplifying issues. They are likely critical of irrelevant or overly optimistic content and prefer answers that reflect a thoughtful understanding of multifaceted topics. Their interests span cultural, social, and practical domains, and they engage with information in a way that prioritizes depth, relevance, and fairness.

Qwen3-32B – User1118 Persona

The synthesized persona is a **curious, practical, and socially engaged individual** who values:
1. **Nuanced, balanced perspectives** on complex topics (e.g., religion, relationships).
2. **Specific, actionable details** in practical contexts (e.g., recipes, travel recommendations).
3. **Relevance and coherence** in responses, avoiding tangents or vague generalizations.
4. **Empathy and community focus**, showing interest in shared experiences (e.g., football culture, fan passion).
5. **Efficiency in communication**, preferring concise yet thorough explanations over fluff.

This persona is likely a **lifelong learner** who enjoys exploring diverse topics but prioritizes clarity and utility in information. They are open to multiple viewpoints but expect responses to acknowledge complexity without oversimplification.