# *Group Preference Alignment*: Customized LLM Response Generation from In-Situ Conversations

**Anonymous ACL submission**


## Abstract

LLMs often fail to meet the specialized needs of distinct user groups due to their `one-size-fits-all` training paradigm (Lucy et al., 2024) and there is limited research on what personalization aspects each group expect. To address these limitations, we propose a group-aware personalization framework, `Group Preference Alignment` (`GPA`), that identifies context-specific variations in conversational preferences across user groups and then steers LLMs to address those preferences. Our approach consists of two steps: (1) Group-Aware Preference Extraction, where maximally divergent user-group preferences are extracted from real-world conversation logs and distilled into interpretable rubrics, and (2) Tailored Response Generation, which leverages these rubrics through two methods: a) `Context-Tuned Inference` (`GPA-CT`), that dynamically adjusts responses via context-dependent prompt instructions, and b) `Rubric-Finetuning Inference` (`GPA-FT`), which uses the rubrics to generate contrastive synthetic data for personalization of group-specific models via alignment. Experiments demonstrate that our framework significantly improves alignment of the output with respect to user preferences and outperforms baseline methods through automated evaluations, while maintaining robust performance on standard benchmarks.


## 1 Introduction

Large Language Models (LLMs) are pivotal in modern natural language processing (NLP), driving applications such as conversational agents, content generation, and automated reasoning (Liu et al., 2024; Tian et al., 2024; Mondal et al., 2024). Despite their remarkable capabilities, LLMs often fall short in addressing the specialized needs of distinct user groups due to their `one-size-fits-all` training paradigm (Lucy et al., 2024). This approach predominantly relies on asking human or LLM judges to provide ratings (e.g. preferred and dispreferred labels) to alternative outputs for the same input query to create *paired* preference data (Ji et al., 2024). These approaches assume that human and AI annotators accurately reflect the preferences of the target user population. Moreover, when models are aligned to this preference data, model outputs will be steered toward the most prevalent preferences of the *annotator* population, even when users express diverse preferences for the same task/query.

Broad preference alignment like this can lead LLMs to produce suboptimal outputs for a *target* user base for two primary reasons. First, the distribution of preferences in the target population may differ from those expressed in the annotator population (e.g., if annotators are generally non-experts within a domain, but the target users are experts). Examples include domain-specific expertise (e.g., a mathematician may struggle with academic writing) and cultural norms (e.g., Japanese audiences may prefer narratives on family bonding, while U.S. audiences favor individualistic themes). Second, even across populations, preference differences may vary with respect to intent/domain (Figure 6 and Table 5). For instance, in education, experts may expect precise terminology and assume foundational knowledge, while novices may desire real-world analogies and step-by-step explanations. In programming, experts often prefer concise debugging strategies, whereas novices may seek explicit concept explanations with visual aids.

Existing methods for group-aware preference adaptation (Balepur et al., 2025; Li et al., 2024a) focus on the first issue only and aim to improve response generation through the use of *personas*. These approaches either use abstract descriptions of personas or auxiliary data reporting general group preferences (e.g., cultural norms) to generate synthetic preference data. However, these methods are limited by the use of external preference signals or internal LLM knowledge of likely prefer-

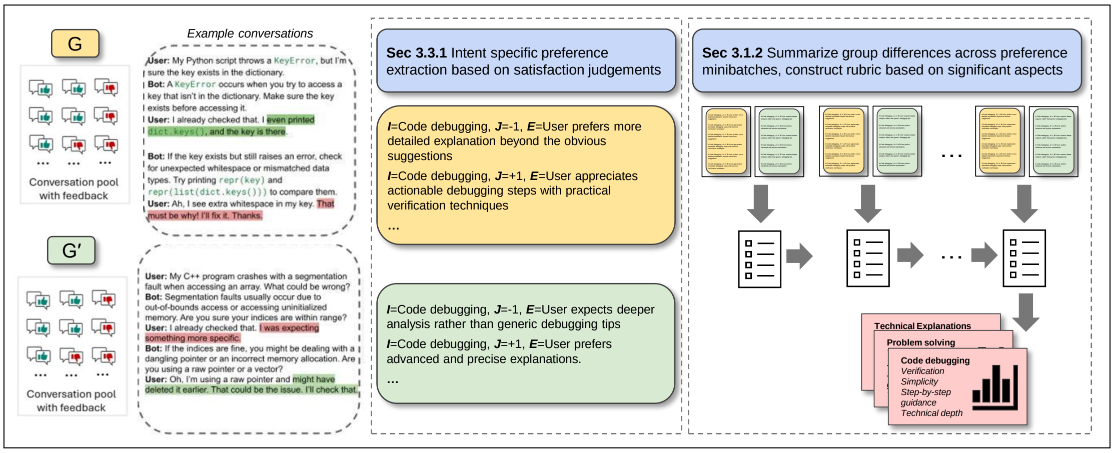


Figure 1: Overview of GPA Rubric Extraction. It Illustrates group aware preference extraction across two groups (*Expert* v. *Novice*) with conversations about Docker and .env file integration. First intent, satisfaction judgments, and individual preferences are extracted from conversations (Sec 3.1 and the extracted preferences are grouped into minibatches and contrasted to extract salient differences across groups and summarize into maximally divergent intent-specific rubrics (Sec 3.1.2).

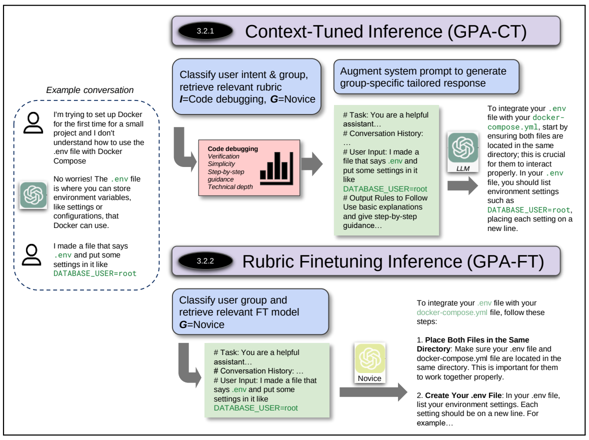


Figure 2: Illustrates tailored response generation for a *Novice* user with a Code debugging intent using `GPA-CT` (Sec 3.2.1) and `GPA-FT` (Sec 3.2.2).

ences for a specified user segment. Specifically, they are unlikely to capture the full range of group-specific preferences expressed across various contexts, which can be directly observed via *in-situ* user interactions. As users prefer responses for varied reasons (Kirk et al., 2024), models tuned on preference data should customize outputs to meet these user-group specific needs (Salemi et al., 2024b; Li et al., 2024a) while also keeping in mind that intent further modulates user preferences. This highlights the need for **group-aware contextual** preference learning to extract users' diverse preferences for tailored response generation.

In this paper, we aim to address this issue by proposing a novel group-aware customization framework, `Group Preference Alignment` (`GPA`), that automatically identifies context-specific variations in conversational preferences across user groups and steers LLMs to address those preferences. Our approach consists of two components. **First**, we propose a method to extract salient group preference differences from real-world conversation logs. We then distill the output into interpretable rubrics that summarize intent-specific guidance for each group (Fig 1). If the groups do not express significant preference differences on specific intents, the returned rubric set will be empty (See 6), hence signifying **no need for customization**. Our method is generalizable across domains/ user groups (See I). **Second**, we propose two methods to use these rubrics to tailor personalized responses for each group: 1) `Context-Tuned Inference` (`GPA-CT`) dynamically adjusts responses via context-dependent in-prompt augmentation which is data-efficient (Figure 7) and training-free, and `Rubric-Finetuning Inference` (`GPA-FT`) uses the learnt rubrics to generate contrastive synthetic data to fine-tune separate models towards group-specific preferences (Fig 2).

Our experiments on two in-situ conversational datasets (Microsoft Copilot logs and WildChat (Zhao et al., 2024)) demonstrate that models customized with `GPA` outperform all baseline methods, including static-preference, persona-guided and zero-shot models, using LLM persona-guided evaluation (Koutcheme et al., 2024; Dong et al., 2024) and improves user satisfaction (Section 5). Notably, alignment with `GPA` produces these improvements without compromising LLM's core capabilities, as evidenced by robust performance on standard benchmarks such as MT-Bench (Zheng et al., 2023) and Arena-Hard (Li et al., 2024b).

## 2 Problem Definition and Notations

We hypothesize that intent-driven user preferences can be automatically extracted from real-world conversation logs between human and AI agents, enabling more effective model alignment than traditional methods that do not incorporate direct user feedback. Consider a user group $\mathcal{G}$ that generates queries for a specific intent $\mathcal{I}$. The responses from the LLM, denoted as $Y_{\mathcal{I}} = \text{LLM}(X_{\mathcal{I}})$, receive user judgments $\mathcal{J}_{\mathcal{G}}(Y_{\mathcal{I}})$ in the form of thumb feedback or implicit textual feedback (eg. thanking the AI). When these preferences diverge from the general population's judgments $\mathcal{J}_P(Y_{\mathcal{I}})$, we hypothesize that aligning the model with group-specific signals will improve response relevance and user satisfaction. Note that if $\mathcal{J}_{\mathcal{G}}(Y_{\mathcal{I}}) \approx \mathcal{J}_P(Y_{\mathcal{I}})$, alignment to $\mathcal{J}_{\mathcal{G}}(Y_{\mathcal{I}})$ will simply reinforce existing preferences in the general population without degrading performance. Unlike RLHF (Ouyang et al., 2022) and RLAIF (Bai et al., 2022), which optimize for majority preferences, our approach leverages in-situ user judgments to achieve fine-grained, group-specific alignment, that is of particular use when user needs deviate significantly from broader norms.

Let $C = \{C_1, C_2, \ldots, C_n\}$ represent a set of conversations from a collection of users, where each $C_i$ is an individual conversation. Let each conversation $C_i$, consisting of $t$ interaction turns of user-agent utterances, be represented as: $C_i = [U_1, A_1, \ldots, U_t, A_t]$. Here, $U_t$ refers to a user utterance and $A_t$ refers to an AI agent response. The user-agent conversations $C_i$ often consist of multiple turns, i.e., $t \geq 1$. Each conversation $C_i$ is labeled with a predicted intent $\mathcal{I}_i$ (see e.g., Wan et al. (2024)). Each conversation turn $U_t$ has been labeled with a user satisfaction judgment $\mathcal{J}_i \in [-1, +1]$ using Lin et al. (2024) and intent $\mathcal{I}_i$ Wan et al. (2024)). Finally, we assume that each user $u$ is associated with one of two groups, ie. $u \in \mathcal{G}$ or $u \in \mathcal{G}'$. Note that in cases where contrasting group labels are unavailable, GPA can also be used by comparing a single group $\mathcal{G}$ against the overall population $P$. We do not make any assumptions about $|\mathcal{G}|$ as long as there are sufficient interactions from users in $\mathcal{G}$ to extract preferences.

## 3 Group Preference Alignment (GPA)

Our GPA framework enables **context-aware, user-group-specific adaptation**, ensuring **more precise and effective model alignment** beyond more conventional preference optimization using auxiliary annotators. The overall approach to align models with in-situ preferences involves two main steps: (i) Generating rubrics with group-aware preference extraction (Section 3.1), and (ii) Tailoring responses based on the extracted rubrics (Section 3.2). We discuss each in more detail below.

### 3.1 Group-Aware Preference Extraction

GPA automatically identifies context-specific variations in conversational preferences across user groups $\mathcal{G}$ and $\mathcal{G}'$ and summarizes the divergent preferences into rubrics (Figure 1 and Algorithm 1). Specifically, given conversations regarding specific intents $\mathcal{I}$ from users in $\mathcal{G}$ and $\mathcal{G}'$, we first extract satisfaction judgments $\mathcal{J}$ from user responses. The algorithm then uses the judgments to infer individual preferences $\mathcal{E}$ that explain the user's positive or negative feedback (Section 3.1.1). Next, the preferences are summarized into generalized preference aspects $\mathcal{A}$, capturing salient differences between two groups (Section 3.1.2). The resulting group-specific rubrics serve as the foundation for group-aware customization.

#### 3.1.1 Extract Intent-Specific Preferences

Algorithm 1 (Lines 1-13) show how we learn group-specific preference rubrics based on user conversations and their corresponding intent labels or context. The input includes a conversation set $C$, user groups $\mathcal{G}$ and $\mathcal{G}'$, intent labels $\mathcal{I}$, a Likert scale threshold $\ell$, and a minibatch size $m$. The algorithm processes each conversation $C_i$ consisting of $t_i$ interaction turns. For each turn $S_j$, the algorithm checks whether the turn expresses implicit satisfaction (SAT) or dissatisfaction (DSAT) judgment through a function $\mathcal{J}(S_j)$. If the turn expresses SAT or DSAT (ie. $abs(\mathcal{J}(S_j)) = 1$), we use an LLM to infer individual preferences and generate an explanation ($\mathcal{E}_+$ or $\mathcal{E}_-$ for SAT and DSAT judgments respectively) [Prompts in 11 and 12]. These explanations, $\mathcal{E}_+$ and $\mathcal{E}_-$, are then grouped by intent $\mathcal{I}_k$ for each user group $\mathcal{G}$ and $\mathcal{G}'$. At the end of this phase, the preferences are organized into sets by user groups, intents and satisfaction.

#### 3.1.2 Summarize Group Differences

Algorithm 1 (Lines 14-30) describe how group-specific preferences are summarized into intent-specific rubrics. For each intent $\mathcal{I}_k \in \mathcal{I}$, the algorithm partitions the preferences of each user group

$(\mathcal{G}, \mathcal{G}')$ into minibatches of size $m$. The algorithm then iterates over pairs of minibatches, one from each group, and summarizes/updates the divergent aspects of their expressed preferences. Specifically, for a pair of minibatches ($\mathcal{E}^a_{G,I_k}$ and $\mathcal{E}^b_{G',I_k}$), the algorithm extracts a set of *aspects* $\mathcal{A}$ that summarize how a preference differs across the two groups (see Figure 1 for illustration). The algorithm also estimates a divergence score $r$ based on Likert scale to rate the significance of each aspect. If the divergence score $r$ exceeds a threshold $\ell$, indicating a significant difference in preferences between the two groups, the aspect $\mathcal{A}_{ab}$ is added to the rubric for the current intent $\mathcal{R}_{I_k}$. Each iteration is provided the aspects from the previous round, so the algorithm can update/refine the aspects as it processes all the minibatches. The process continues until all significant divergent aspects have been identified and included in $\mathcal{R}_{\mathcal{I}_k}$. Finally, the rubric for this intent is added to the global rubric list $\mathcal{R}$. The algorithm returns the full set of rubrics $\mathcal{R}$ and an interpretation of each rubric. These capture the distinct preference patterns of the two user groups across intents.

## 3.2 Response Tailoring

After learning rubrics in section 3.1, we outline the following two methods to align LLMs based on these learnt rubrics/preferences. The first method, `GPA-CT`, involves dynamically augmenting prompts to produce group-aware tailored responses. It dynamically adjusts the LLM prompts incorporating learnt rubrics from 3.1 during inference, conditioned on intent and user group identified for each conversation (Section 3.2.1). The second approach, `GPA-FT` uses learnt rubrics to synthetically augment conversational data with paired responses that reflect group-specific preferences conditioned on specific intents. This process produces a tailored set of preference data. We finetune a group-specific LLM using this enriched dataset to enhance their alignment to the targeted group (Section 3.2.2).

### 3.2.1 `GPA-CT`: Dynamic Context-Tuning

Context-tuning with `GPA-CT` is an adaptive process that infers the user's group and intent, retrieves the relevant rubrics for that intent, and then modifies the instructions sent to the LLM to generate the next output (Algorithm 2 in Appendix B and Figure 2). Unlike finetuning, which adjusts a model's weights based on a fixed training dataset, context-tuning allows for dynamic adjustments to the model's prompt based on real-time analysis of user intent and group membership. This means the model can adapt to the specific needs of different user groups on-the-fly without requiring specialized group-specific models. `GPA-CT` offers several advantages over finetuning, including the flexibility to adapt to user-specific needs without retraining and enhanced efficiency as it avoids the extensive resources typically required for finetuning.

### 3.2.2 `GPA-FT`: Rubric-Guided Contrastive Data Generation and Fine-tuning

Rather than merely fine-tuning LLMs with the training data comprising of preference signals from user groups ($\mathcal{G}$ and $\mathcal{G}'$), we use our learned rubrics to generate more realistic contrastive pairs that vary according to observed preference dimensions. `GPA-FT` may be favored over `GPA-CT` in situations where the LLM is less steerable with prompt-tuning (eg. smaller models) and/or when lower latency is desired. Specifically, we finetune LLMs using synthetic training data generated with intent- and group-aware rubrics to reflect in-situ user preferences. Algorithm 4 (Appendix B) describes our approach to rubric-guided data generation. It takes as input our dataset of conversations and augments the existing AI responses with paired responses of opposing preference polarity, conditioned on the group-aware rubrics. Consider a conversation $S_i$ up to the $j^{th}$ user utterance, with $A_j$ referring to the corresponding AI response, and $\mathcal{J}(S_i) \in \{+1, -1\}$ referring to the user satisfaction judgment for $A_j$. To generate *contrastive augmented samples*, we modify responses as follows: If $\mathcal{J}(S_i) = +1$ (preferred response), we generate a *dispreferred response* $A_{aug}$ by instructing the LLM to incorporate features from the opposing group's rubric for that intent. Otherwise, if $J(S_i) = -1$ (dispreferred response), we generate a *preferred response* $A_{aug}$ by instructing the LLM to align the output with the user's group rubric for that intent. When applied to the full training data, the procedure produces an augmented dataset $\mathcal{D}_{aug}$ where each original instance is paired with a contrastive sample: $T_{aug} = (S_i,\ A_{aug},\ -\mathcal{J}(S_i))$. Note that $\mathcal{J}_{aug} = -\mathcal{J}(S_i)$ ensures contrastive preference learning. Next, we train separate models for each user group. Given a prompt $S$ and responses $A_+$ (preferred) and $A_-$ (dispreferred), the likelihood of selecting the preferred response is modeled as:

$$P_\theta(A_+|S) = \frac{e^{f_\theta(S,A_+)}}{e^{f_\theta(S,A_+)} + e^{f_\theta(S,A_-)}} \qquad (1)$$

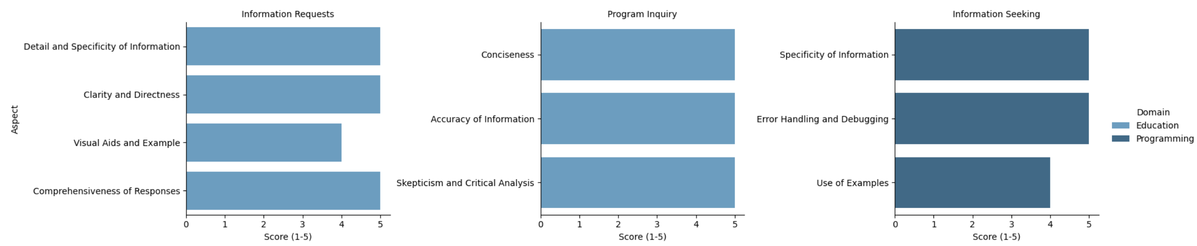


Figure 3: Rubrics/Aspects on which Experts and Novices Differ in the Education and Programming Domains as extracted from BingChat Dataset on 3 intents (Information Requests, Program Inquiry, Information Seeking) with the Likert-Scale Rating (1-5) on the x-axis and aspect/rubric names on the y-axis.

where $f_\theta(S, A)$ is a scoring function parameterized by $\theta$, representing the model's preference alignment. The DPO objective is to maximize the log-likelihood of the chosen response:

$$\mathcal{L}_{DPO} = \mathbb{E}_{(S,A_+,A_-)\sim\mathcal{D}_{aug}} \left[\log P_\theta(A_+|S)\right] \quad (2)$$

By optimizing $\mathcal{L}_{DPO}$, the model learns to **prefer responses aligned with group-specific rubrics** while discouraging responses reflecting dispreferred aspects. We train two specialized DPO models ($P_{\theta_G}$ and $P_{\theta_{G'}}$) using contrastive samples from $\mathcal{D}_{aug}$ for each (See Algorithm 5, Appendix B) for how we apply the finetuned models for inference). This results in: **[1]** $P_{\theta_G}$, optimized to generate responses aligned with the preferences of user group $G$. **[2]** $P_{\theta_{G'}}$, optimized for user group $G'$ [1].

## 4 Experimental Setup

We evaluate GPA using real-world conversational logs from Microsoft Copilot and Wildchat (Zhao et al. (2024)) data. For the intent category *programming and software*, we use 8000 Copilot conversations and 8200 WildChat conversations. We group conversations into *expert* (i.e., $\mathcal{G}$; Copilot: 2200, WildChat: 6000) and *novice* (i.e., $\mathcal{G}'$; Copilot: 5800, WildChat: 2000) groups by using an auxiliary *expertise* classifier [2], (Copilot: 5800 novice, 2200 expert; WildChat: 6000 expert, 2000 novice). Next, we consider the intent category *Creative writing and editing* in WildChat and form user groups based on metadata, specifically *location*, partitioning users into USA (8000 conversations) and China (800 conversations) (Full Dataset Statistics in Appendix). We partitioned the above datasets into 90:10 train:test split to ensure no training signal leakage. Next, we use predicted SAT/DSAT judgments to learn divergent preferences on the training data. Following Lin et al. (2024) and the taxonomy proposed by Shi et al. (2024), we used GPT-4o to classify bot responses resulting in a subsequent user SAT, DSAT, or Neither judgment. We fine-tune using synthetic data constructed from the full training set.

**Models and GPA Baselines.** For rubric extraction, we use GPT-4o and for tailored response generation (GPA-CT and GPA-FT), we use two base LLMs ($M$): `gemma-2-9b-it`[3] (Team et al., 2024) and `Meta-Llama-3-8B`[4] (Grattafiori et al., 2024). We compare GPA-CT and GPA-FT against several baselines: a) **Zero-shot (Base)** responses, b) **Persona-Aware (Persona-P)**: which augments the input prompt with persona (P) information to mimic responses from specific user-groups through role-playing behavior, (Prompt 10) c) **Persona-Criteria-Aware (Static-P)**: which uses $M$ to first generate preference criteria for $G$ and $G'$, and then append the generated criteria to the prompt (Prompt 9), d) **KTO (KTO-P)**: which fine-tunes an LLM with SAT and DSAT samples to tailor towards each persona using KTO (Ethayarajh et al., 2024), e) **KTO-Augmented (KTO-P')**: which also uses KTO to finetune an LLM on the SAT and DSAT samples from each persona, this time augmented with the contrastive pairs generated by rubrics.

**Evaluation Metrics.** We evaluate responses across three key dimensions: **1) Customization to User-Group Preferences**: We assess alignment with group-specific preferences using Win-Tie-Lose (WTR) rates computed via GPT-4o-as-a-Judge with Persona-Role Playing (Dong et al., 2024) (Prompt 18). We also report WTR results for

[1]We also experiment with KTO but found DPO to be superior (Appendix H)

[2]We manually inspected 100 random conversations and found that the classification was reliable ($\kappa = 0.88$ agreement computed between the first author and GPT-4o).

[3]https://huggingface.co/google/gemma-2-9b-it

[4]https://huggingface.co/meta-llama/Meta-Llama-3-8B

| Model | LLM Pref (W/L/T) | LLM conf $\geq 75$ | LLM Pref (W/L/T) | LLM conf $\geq 75$ |
|---|---|---|---|---|
| | **Intent=Programming/Group=Novice** | | **Intent=Programming/Group=Expert** | |
| GPA-CT vs Base | 65.82 / 25.00 / 9.18 | 67.53 / 32.47 | 57.10 / 42.04 / 0.86 | 57.46 / 42.54 |
| GPA-CT vs Persona | 60.44 / 31.96 / 7.60 | 73.97 / 26.3 | 61.10 / 38.30 / 0.6 | 61.91 / 38.09 |
| GPA-CT vs Static | 56.43 / 37.43 / 6.14 | 80.00 / 20.00 | 57.38 / 41.47 / 1.6 | 59.05 / 40.95 |
| GPA-FT vs Base | 71.29 / 25.87 / 2.84 | 68.05 / 31.95 | 53.17/ 40.62 / 5.56 | 56.15 / 43.84 |
| GPA-FT vs Persona | 70.98 / 27.76 / 1.26 | 68.84 / 31.16 | 58.80 / 40.62 / 5.0 | 59.62 / 40.37 |
| GPA-FT vs Static | 66.88 / 32.18 / 0.95 | 60.64 / 39.36 | 59.65 / 39.77 / 0.56 | 57.72 / 42.27 |
| GPA-FT vs GPA-CT | 63.09 / 36.59 / 0.32 | 57.59 / 42.41 | 53.12 / 38.35 / 0.28 | 58.99 / 41.00 |
| | **Intent=Writing/Group=USA** | | **Intent=Writing/Group=China** | |
| GPA-CT vs Base | 45.5 / 53.5 / 1.0 | 54.1 / 45.9 | 58.5 / 23.9 / 17.6 | 88.57 / 11.42 |
| GPA-CT vs Persona | 55.5 / 42.5 / 2.0 | 59.5 / 40.5 | 53.6 / 28.73 / 17.60 | 60.0 / 40.00 |
| GPA-CT vs Static | 67.02 / 31.00 / 1.98 | 67.10 / 32.90 | 52.11 / 32.3 / 15.59 | 68.57 / 31.43 |
| GPA-FT vs Base | 55 / 26.5 / 18.5 | 62.2 / 37.8 | 55.22 / 20.84 / 23.94 | 60.95 / 39.04 |
| GPA-FT vs Persona | 77 / 21.5 / 1.5 | 82.4 / 17.5 | 35.21 / 40.84 / 23.95 | 32.38 / 67.62 |
| GPA-FT vs Static | 85 / 14.5 / 0.5 | 88.5 / 11.5 | 28.16 / 54.92 / 16.92 | 47.61 / 52.39 |
| GPA-FT vs GPA-CT | 85.5 / 14 / 0.5 | 71.4 / 28.6 | 39.43 / 40.84 / 19.73 | 40.95 / 59.05 |

Table 1: Table showing the WR Evaluation of GPA-CT and GPA-FT on Wildchat Creative Writing Domain across countries/culture, and Bing Dataset on Programming Domain. LLM conf=Expected Confidence (Dong et al., 2024), W/L/T=win/lose/tie rates of our methods against baselines.

Confidence-Estimated LLMs (Dong et al., 2024) at a confidence threshold correlated with human judgment. To mitigate positional bias, we average win rates by swapping response positions. **2) Oracle-Guided Satisfaction Estimation**: We identify responses that significantly deviate from DSAT-classified reference responses to minimize negative follow-up feedback. This evaluation measures whether our methods generate fewer dissatisfactory signals than baselines, focusing solely on response differences without considering user personas (Prompt 16). Success is determined by the number of times our responses outperform baselines in WTR comparisons. **3) Quality Evaluation on Standard Benchmarks**: We assess the generalization of GPA-CT and GPA-FT using two open-ended instruction-following benchmarks: MT-Bench (Zheng et al., 2023) and Arena-Hard (Li et al., 2024b). This ensures that our models maintain strong performance in general instruction-following tasks despite group-aware alignment. We follow each benchmark's evaluation protocol, reporting Win Rate (WR) for Arena-Hard (with GPT-4o as the judge) and the average MT-Bench score using default inference strategies.

| Model | WR | LR | TR | $\Delta$ (%) | MT-Bench |
|---|---|---|---|---|---|
| Base | | | | | 8.320 |
| Novice GPA-FT | 49.11 | 39.10 | 8.62 | +10.01 | 8.334 |
| Expert GPA-FT | 47.89 | 42.88 | 6.41 | +5.01 | 8.212 |
| US GPA-FT | 47.56 | 43.80 | 8.64 | +3.76 | 8.256 |
| China GPA-FT | 48.49 | 38.34 | 9.02 | + 10.15 | 8.300 |

Table 2: Comparison against LLama Base on Arena-Hard Benchmark (Win/Lose/Tie, and Win-Lose $\Delta$) and evaluation on MT-Bench.

| Setup | Win (%) | Lose (%) | Tie (%) |
|---|---|---|---|
| GPA-CT vs Base | 69.61% | 29.41% | 0.98% |
| GPA-CT vs Persona | 65.69% | 33.33% | 0.98% |
| GPA-CT vs Static | 76.70% | 21.36% | 1.94% |

Table 3: WTR against the baselines (Win determines the number of times GPA-CT is chosen over others) on Wildchat Programming when compared against reference DSAT Evaluation using Llama.

## 5 Main Results and Findings

**GPA-FT excels when ample finetuning data is available, while GPA-CT remains robust in lower-data settings.** GPA-FT using Llama consistently outperforms all baselines when sufficient finetuning data is available, particularly in the Novice and US groups (Table 1). In the Novice category, GPA-FT achieves a 71.29% Win Rate (WR) vs. Base, compared to 65.82% for GPA-CT, showing that finetuning helps models better adapt to novice preferences. A similar trend is observed in the US Writing group, where GPA-FT achieves an 85% WR vs. Static, outperforming GPA-CT at 67.02% WR. Conversely, GPA-CT is more effective when data is scarce (also evident in Figure 7), as seen in China and Expert groups. Against the Base model, GPA-FT achieves only a 55.22% WR in China, while GPA-CT performs slightly better at 58.5% WR, indicating that in-context adaptation is more useful in this setting. These observations also hold on the Wildchat Dataset where we observe a drop in the WTR of GPA-FT compared with GPA-CT (Table 4), since the Novice Groups have correspondingly fewer samples.

**Base models such as Llama/Gemma are tuned more towards US Preferences and Expert Groups compared to China/Novice Groups.** Table 1 shows that for Expert groups, Base is the hardest baseline to beat, reinforcing the fact that pretrained models are already aligned with expert preferences. For example, GPA-FT vs. Base in Expert groups achieves only a 53.12% WR, compared to 71.29% WR for Novice users, indicating that Base already reflects expert-style responses well. The same trend appears in US-based writing conversations, where GPA-CT and GPA-FT struggle more against the Base model than against Persona or Static baselines, proving that pretraining biases favor Western-aligned outputs. Similar observations also hold true when Gemma is used as the base model (Table 19).

**GPA-CT improves overall satisfaction compared to baselines.** Results in Table 3 (computed on the DSAT Signals in the Wildchat Programming Domain) using Llama show that GPA-CT consistently wins against all setups, achieving the highest win rate against Static (76.70%), followed by Base (69.61%) and Persona (65.69%). This suggests that GPA-CT generates responses that better align with user expectations and reduces dissatisfaction signals when compared with other baselines.

**GPA-FT does not compromise model performance on other benchmarks.** We evaluate our LLama-based GPA-FT models on MT-Bench and Arena-Hard to assess if their instruction-following performance degrades on standard benchmarks as done by Shi et al. (2024). Table 2 confirms that GPA-FT does not compromise performance on the instruction-following benchmarks. The MT-Bench scores for GPA-FT models remain close to the Base model (8.320), with Novice GPA-FT (8.334) even slightly outperforming it. Similarly, Arena-Hard results show a positive win-loss delta ($\Delta$) across all groups, with China GPA-FT (+10.15%) and Novice GPA-FT (+10.01%) achieving the highest gains. Even in Expert (+5.01%) and US (+3.76%) categories, GPA-FT maintains competitive performance.

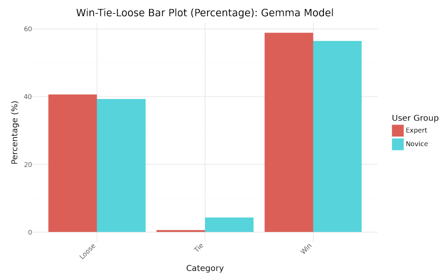


Figure 4: Bar plot evaluating Gemma outputs from intent-aware rubric creation vs intent-unaware rubric creation using GPA-CT. Results shows that intent/context heavily impacts performance when GPA approach is used to personalize responses on BingChat Test Set.

Overall these findings reinforce the fact that group preference alignment via fine-tuning does not lead to overfitting or loss of generalization.

## 6 Further Analysis

**User preferences vary across cultures and domains.** Cultural background significantly shapes user preferences, even for the same intent. US users prefer personal engagement and detailed narratives, while Chinese users favor clarity and structured summaries in writing and creative content (Table 5). Similarly, when expertise remains constant but the domain shifts from Education to Programming, experts prioritize different aspects—educators value comprehensiveness and critical analysis, whereas programmers focus on specificity, error handling, and debugging (Figure 3).

**Intent-Specific Rubrics are important for better group-preference alignment compared to generic ones.** To investigate the impact of intents in preference learning, we extracted rubrics from the Bing expertise groups in two ways: without considering intent and with intent-awareness. These rubrics were then used for context-tuning on a held-out test set, followed by WTR evaluation using Persona-based evaluation (Appendix 18). The results show a notable drop in WR when intent was not used (Figure 4), demonstrating that intent-aware rubric extraction leads to more personalized, contextually aligned responses.

**Preference Rubrics degrade significantly when expertise labels are randomly flipped, indicating robustness.** We assess the robustness of our extracted rubrics by randomly flipping expertise labels and extracting out the preference rubrics using GPA. Figure 5 highlights the impact of random

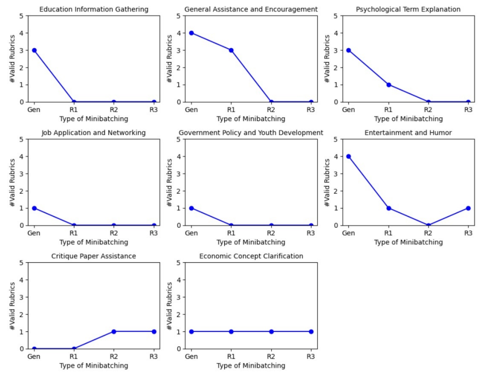


Figure 5: Illustrates how random shuffling of expertise labels impacts rubric generation. It reveals that, for most intents, shuffling results in the extraction of predominantly invalid rubric items, ultimately reducing the overall quality and number of valid rubric extractions.

shuffling of expertise labels on rubric generation across various intents, where the validity of generated rubrics is determined by a self-correcting evaluation prompt from GPA. The results demonstrate that valid rubric generation is most successful under the original generation strategy (Gen) with correctly aligned expertise labels. However, as expertise labels are randomly shuffled (R1, R2, R3), the number of valid rubrics decreases significantly, often to zero, which confirm the robustness of our extracted rubrics.

## 7 Related Work

Customization of user interactions to better serve both individual and group preferences has a long history of research in a range of fields that leverage language technology. These include recommender systems (Cho et al., 2002; Zhou et al., 2012), search and information retrieval (Teevan et al., 2005; Tabrizi et al., 2018), education (McHugh et al., 2020; Domenichini et al., 2024), and healthcare (Wang et al., 2020; Li et al., 2024d).

Meanwhile, LLMs are trained in a `one-size-fits-all` paradigm (Lucy et al., 2024) where large-scale ratings from auxiliary human annotators or LLMs in a paired preference setup is used to teach models to generate preferred responses. This can make them difficult to customize. Nevertheless, recent work (Zeng et al., 2023; Sorensen et al., 2024) has begun to advocate for the need for LLMs to serve more diverse preferences through pluralistic alignment. Much of the work in LLM customization has focused on personalizing systems to the individual (Kirk et al., 2024; Salemi et al., 2024b). These have used a variety of different approaches, including retrieval-augmented generation (Salemi et al., 2024a), memory (Zhang et al., 2023), parameter-efficient fine-tuning (Tan et al., 2024), and reinforcement learning (Poddar et al., 2024). Personalized LLM systems have also been applied to diverse applications, such as contextual query suggestion (Baek et al., 2024) and document creation (Mondal et al., 2024).

Recently some attempts have been made at modeling a large number of individual characteristics at scale, such as with a thousand preferences (Lee et al., 2025) or a million personas (Ge et al., 2024). However, the focus on modeling *group* preferences has been limited to a few recent research efforts (Feng et al., 2024; Zhao et al., 2023; Ramesh et al., 2024). Crucially, none of these methods leverage real-world conversational data at scale to *learn* these group preferences. While some recent work has begun to incorporate feedback from in-situ user-AI interactions in order to improve models (Shi et al., 2024; Li et al., 2024c), their focus has been different from modeling group preferences. Thus, to the best of our knowledge, our paper is the first attempt at using large-scale satisfaction signals from human-AI conversation logs to customize LLM responses with group preference alignment (More in Appendix A).

## 8 Conclusion

In this work we address a critical gap in group-aware personalization of LLMs by developing our Group Preference Alignment (GPA) framework. This framework identifies and incorporates diverse conversational preferences of distinct user groups via a two-step process of Group-Aware Preference Extraction and Tailored Response Generation. Our experiments demonstrate that GPA significantly enhances the alignment of LLM outputs with group-specific preferences. GPA outperforms baseline methods with respect to preferences while maintaining robust performance on the standard information-following benchmarks. This work paves the way for developing more personalized and contextually aware LLMs using *in-situ* interactions using interpretable rubrics, which will ultimately improve user satisfaction and engagement. Due to increased transparency, this approach can be scaled up in legal/healthcare and other such high-stake domains.

## Limitations

While Group Preference Alignment (GPA) framework significantly improves user-group-specific response alignment, it has a few limitations:

**a) Dependence on Predefined User Groups**: The effectiveness of GPA relies on the availability of well-defined user groups with sufficient interaction data. In cases where user preferences are highly individualized or overlap across groups, extracting meaningful rubrics becomes challenging.

**b) Scalability of Rubric Extraction**: The framework extracts group-specific rubrics from conversation logs, which can be computationally expensive for large datasets. Additionally, the process assumes that conversational preferences remain stable within each group, which may not always be the case.

**c) Contextual Drift Over Time**: User preferences evolve, especially in dynamic domains like technology and education. The extracted rubrics may become outdated, requiring periodic updates to maintain alignment with current user expectations.

**d) Applicability Across Domains**: While GPA is tested on education, programming, and writing domains, its generalization to other highly specialized fields (e.g., legal or medical domains) remains unexplored. Future work should assess its adaptability to such contexts.

## References


Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. 2024. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM on Web Conference 2024*, pages 3355–3366.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Nishant Balepur, Vishakh Padmakumar, Fumeng Yang, Shi Feng, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2025. Whose boat does it float? improving personalization in preference tuning via inferred user personas. *Preprint*, arXiv:2501.11549.

Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. 2002. A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications*, 23(3):329–342.

Diana Domenichini, Filippo Chiarello, Vito Giordano, and Gualtiero Fantoni. 2024. Llms for knowledge modeling: Nlp approach to constructing user knowledge models for personalized education. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 576–583.

Yijiang River Dong, Tiancheng Hu, and Nigel Collier. 2024. Can LLM be a personalized judge? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10126–10141, Miami, Florida, USA. Association for Computational Linguistics.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *Preprint*, arXiv:2402.01306.

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. *arXiv preprint arXiv:2406.15951*.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *Preprint*, arXiv:2406.20094.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park,

Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal

Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. *arXiv preprint arXiv:2306.01200*.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. 2024. Ai alignment: A comprehensive survey. *Preprint*, arXiv:2310.19852.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A. Hale. 2024. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nat. Mac. Intell.*, 6:383–392.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520*.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators. *arXiv preprint arXiv:2309.17012*.

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024. Open source language models can provide feedback: Evaluating llms' ability to help students using gpt-4-as-a-judge. *Preprint*, arXiv:2405.05253.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. 2025. Aligning to thousands of preferences via system message generalization. *Advances in Neural Information Processing Systems*, 37:73783–73829.

Cheng Li, Mengzhou Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024a. Culturellm: Incorporating cultural differences into large language models. *Preprint*, arXiv:2402.10946.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024b. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *Preprint*, arXiv:2406.11939.

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

Yu-Hao Li, Yu-Lin Li, Mu-Yang Wei, and Guang-Yu Li. 2024d. Innovation and challenges of artificial intelligence technology in personalized healthcare. *Scientific reports*, 14(1):18994.

Ying-Chun Lin, Jennifer Neville, Jack Stokes, Longqi Yang, Tara Safavi, Mengting Wan, Scott Counts, Siddharth Suri, Reid Andersen, Xiaofeng Xu, Deepak Gupta, Sujay Kumar Jauhar, Xia Song, Georg Buscher, Saurabh Tiwary, Brent Hecht, and Jaime Teevan. 2024. Interpretable user satisfaction estimation for conversational systems with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11100–11115, Bangkok, Thailand. Association for Computational Linguistics.

Na Liu, Liangyu Chen, Xiaoyu Tian, Wei Zou, Kaijiang Chen, and Ming Cui. 2024. From llm to conversational agent: A memory enhanced architecture with fine-tuning of large language models. *Preprint*, arXiv:2401.02777.

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2023. Llms as narcissistic evaluators: When ego inflates evaluation scores. *arXiv preprint arXiv:2311.09766*.

Li Lucy, Su Lin Blodgett, Milad Shokouhi, Hanna Wallach, and Alexandra Olteanu. 2024. "one-size-fits-all"? examining expectations around what constitute "fair" or "good" NLG system behaviors. 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 1054–1089, Mexico City, Mexico. Association for Computational Linguistics.

David McHugh, Sarah Shaw, Travis R Moore, Leafia Zi Ye, Philip Romero-Masters, and Richard Halverson. 2020. Uncovering themes in personalized learning: Using natural language processing to analyze school interviews. *Journal of Research on Technology in Education*, 52(3):391–402.

Ishani Mondal, Shwetha S, Anandhavelu Natarajan, Aparna Garimella, Sambaran Bandyopadhyay, and Jordan Boyd-Graber. 2024. Presentations by the humans and for the humans: Harnessing LLMs for generating persona-aware slides from documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2664–2684, St. Julian's, Malta. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

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

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. 2024. Group robust preference optimization in reward-free rlhf. *arXiv preprint arXiv:2405.20304*.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 752–762.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024b. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, Zexue He, Mengting Wan, Pei Zhou, Sujay Jauhar, Xiaofeng Xu, Xia Song, and Jennifer Neville. 2024. Wildfeedback: Aligning llms with in-situ user interactions and feedback. *Preprint*, arXiv:2408.15549.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. 2024. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*.

Shayan A Tabrizi, Azadeh Shakery, Hamed Zamani, and Mohammad Ali Tavallaei. 2018. Person: Personalized information retrieval evaluation based on citation networks. *Information Processing & Management*, 54(4):630–656.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024. Democratizing large language models via personalized parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.04401*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Jaime Teevan, Susan T Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5836–5847, New York, NY, USA. Association for Computing Machinery.

Zheyu Wang, Haoce Huang, Liping Cui, Juan Chen, Jiye An, Huilong Duan, Huiqing Ge, Ning Deng, et al. 2020. Using natural language processing techniques to provide personalized educational materials for chronic disease patients in china: development and assessment of a knowledge-based health recommender system. *JMIR medical informatics*, 8(4):e17642.

Dun Zeng, Yong Dai, Pengyu Cheng, Tianhao Hu, Wanshun Chen, Nan Du, and Zenglin Xu. 2023. On diverse preferences for large language model alignment. *arXiv preprint arXiv:2312.07401*.

Kai Zhang, Fubang Zhao, Yangyang Kang, and Xiaozhong Liu. 2023. Memory-augmented llm personalization with short-and long-term memory coordination. *arXiv preprint arXiv:2309.11696*.

Siyan Zhao, John Dang, and Aditya Grover. 2023. Group preference optimization: Few-shot alignment of large language models. *arXiv 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. *Preprint*, arXiv:2405.01470.

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.

Xujuan Zhou, Yue Xu, Yuefeng Li, Audun Josang, and Clive Cox. 2012. The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review*, 37:119–132.

# A Additional Related Work

In this paper, we use LLMs as evaluators to measure the quality of system generations. Despite prior work pointing to some pitfalls with this approach, such as bias (Koo et al., 2023) and preferential scoring (Liu et al., 2023), using LLMs with judicious prompting for evaluation of language and information systems has become common practice (Zheng et al., 2023; Koutcheme et al., 2024). Recent efforts have applied the LLM-as-a-judge paradigm to evaluating a variety of applications such as translation (Kocmi and Federmann, 2023) and summarization (Jain et al., 2023); notably these also include personalization (Dong et al., 2024).

# B Algorithm Pseudocode

This appendix summarizes the pseudocode for `GPA-CT` and `GPA-FT` methods. Algorithm 1 depicts the procedure for extracting the group-aware preference rubric which is used in both methods. Inference for `GPA-CT` is next summarized in Algorithm 2. For `GPA-FT`, we next describe the training procedure in Algorithm 3, and the pseudocode for generating the augmentented training examples for finetuning using the rubric is shown in Algorithm 4. Finally, Algorithm 5 applies the fine tuned models for `GPA-FT` inference. We simply need to look up the appropriate group-aware model to use for generation.

# C Intent Creation

The process of creating sub-intents within a domain from user conversations involves an iterative clustering approach. This method refines the set of intents through successive iterations, ensuring that each domain's sub-intents are optimally defined based on the conversation data.

Let $C = \{C_1, C_2, \ldots, C_n\}$ represent the entire conversation history, where each $C_j$ is an individual conversation. First, we prompt each conversation $C_j$ to identify its intent. Let $I_j$ denote the intent of conversation $C_j$. The intent extraction function can be represented as: $I_j = \text{Intent}(C_j)$. This step generates a set of intents $I = \{I_1, I_2, \ldots, I_n\}$ corresponding to each conversation in the pool. Next, the entire conversation pool $C$ is divided into $m$ mini-batches $B_1, B_2, \ldots, B_m$, where each mini-batch $B_k$ contains a subset of conversations: $B_k = \{C_{k1}, C_{k2}, \ldots, C_{kl}\}$ for $k = 1, 2, \ldots, m$. For each mini-batch $B_k$, we use the intents $I_{k1}, I_{k2}, \ldots, I_{kl}$ associated with the

| Model | Novice | Expert |
|---|---|---|
| GPA-CT vs Base | 84.3 / 15.6 | 65.77 / 34.22 |
| GPA-CT vs Persona | 73.4 / 26.5 | 61.74 / 38.25 |
| GPA-CT vs Static | 76.5 / 23.4 | 55.03 / 44.91 |
| GPA-FT vs Base | 64.51 / 35.48 | 54.12 / 45.88 |
| GPA-FT vs Persona | 51.62 / 48.38 | 63.54 / 36.45 |
| GPA-FT vs Static | 45.12 / 54.83 | 58.86 / 41.13 |
| GPA-FT vs GPA-CT | 44.22 / 55.78 | 56.16 / 43.84 |

Table 4: Results for Wildchat with LLama. LLM expected confidence (EC) [LLM conf $\geq$ 75] W/L/T=win/lose/tie

**Algorithm 1** Group-Aware Preference Extraction

**Require:** Conversation set $\mathbf{C}$; User groups $\mathcal{G}$ and $\mathcal{G}'$; Intent labels $\mathcal{I}$
**Require:** Likert scale threshold $\ell$; Minibatch size $m$
**Ensure:** Rubric $\mathcal{R}$

**Step 1: Preference Judgement and Explanation Extraction**
$\mathcal{E}_+ = []; \mathcal{E}_- = []$
**for** each conversation $C_i \in \mathbf{C}$ with $t_i$ turns **do**
  **for** $j = [1..t_i]$ **do**
    $S_j = [U_1, A_1, \ldots, U_j]_{C_i}$
    # If $t_j$ contains judgment, extract pref expectation
    **if** $\mathcal{J}(S_j) == +1$ **then**
      $\mathcal{E}_+ = \mathcal{E}_+ \cup$ { `LLM.InferUserExpectation` $(S_j, \mathcal{J}(S_j))$}
    **if** $\mathcal{J}(S_j) == -1$ **then**
      $\mathcal{E}_- = \mathcal{E}_- \cup$ { `LLM.InferUserExpectation` $(S_j, \mathcal{J}(S_j))$}
# Group prefs $\mathcal{E}_+$ and $\mathcal{E}_-$ by intent $I_k$ for each group
$\mathcal{E}_{\mathcal{G},\mathcal{I}_k} = \{\mathcal{E}_+ \mid C_i \in \mathcal{G}, C_i \text{ matches } \mathcal{I}_k\} \cup \{\mathcal{E}_- \mid C_i \in \mathcal{G}, C_i \text{ matches } \mathcal{I}_k\}$
$\mathcal{E}_{\mathcal{G}',\mathcal{I}_k} = \{\mathcal{E}_+ \mid C_i \in \mathcal{G}', C_i \text{ matches } \mathcal{I}_k\} \cup \{\mathcal{E}_- \mid C_i \in \mathcal{G}', C_i \text{ matches } \mathcal{I}_k\}$
**Step 2: Aspect-Based Rubric Construction**
# Initialize an empty list of rubric items
$\mathcal{R} = []$
**for** each intent $\mathcal{I}_k \in \mathcal{I}$ **do**
  # Uniformly partition each explanation set into minibatches
  $\mathcal{E}_{\mathcal{G},\mathcal{I}_k} = \{\mathcal{E}^1_{\mathcal{G},\mathcal{I}_k}, \ldots, \mathcal{E}^{n_1}_{\mathcal{G},\mathcal{I}_k}\}$ s.t. $\forall a \; |\mathcal{E}^a_{\mathcal{G},\mathcal{I}_k}| = m$
  $\mathcal{E}_{\mathcal{G}',\mathcal{I}_k} = \{\mathcal{E}^1_{\mathcal{G}',\mathcal{I}_k}, \ldots, \mathcal{E}^{n_2}_{\mathcal{G}',\mathcal{I}_k}\}$ s.t. $\forall b \; |\mathcal{E}^b_{\mathcal{G}',\mathcal{I}_k}| = m$
  $\mathcal{A}_{\mathcal{I}_k} = []; r_{\mathcal{I}_k} = \{\}$
  **for** each pair of minibatches $(\mathcal{E}^a_{\mathcal{G},\mathcal{I}_k}, \mathcal{E}^b_{\mathcal{G}',\mathcal{I}_k})$ **do**
    # Extract/update divergent aspects $\mathcal{A}$
    # Score group divergence on Likert scale $r$
    $[\mathcal{A}_{ab}, r_{ab}]$ = `LLM.ExtractAspectsAndLikert` $(\mathcal{E}^a_{\mathcal{G},\mathcal{I}_k}, \mathcal{E}^b_{\mathcal{G}',\mathcal{I}_k}, \mathcal{A}_{\mathcal{I}_k})$
    $\mathcal{A}_{\mathcal{I}_k} = \mathcal{A}_{ab}; r_{\mathcal{I}_k}[\mathcal{A}_{ab}] = r_{ab}$
  $\mathcal{R}_{\mathcal{I}_k} = []$
  **for** each aspect $\mathcal{A}_k \in \mathcal{A}_{\mathcal{I}_k}$ **do**
    **if** $r_{\mathcal{I}_k}[\mathcal{A}_k] > \ell$ **then**
      $\mathcal{R}_{\mathcal{I}_k} \leftarrow \mathcal{R}_{\mathcal{I}_k} \cup \{\mathcal{A}_k\}$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{R}_{\mathcal{I}_k}\}$
**return** $\mathcal{R}$

**Algorithm 2** GPA-CT: Inference

**Require:** Partial conversation $S_i = [U_1, A_1, \ldots, U_j]$ up to $j^{th}$ user utterance
**Require:** Rubric $\mathcal{R}$
**Ensure:** LLM answer $A_j$

**Step 1: Classify user group and intent**
$\mathcal{I}_i$ = `Intent`$(S_i)$
$\mathcal{G}_i$ = `Group`$(S_i)$
**Step 2: Retrieve Rubric and Augment Prompt**
$\mathcal{R}_i = \mathcal{R}_{\mathcal{I}_i}$
$A_j$ = `LLM.ModifyPrompt`$(S_i, \mathcal{G}_i, \mathcal{R}_i)$
**return** $A_j$

conversations in that mini-batch to perform intent clustering. Let $\mathcal{C}_k$ represent the set of

**Algorithm 3** GPA-FT Training

**Require:** Training conversation set **C**
**Require:** $\mathcal{J}$ are the user's preference judgements for the AI turn-level responses $A$
**Require:** Rubric $\mathcal{R}$
**Require:** $Model_{Base}$ is the base LLM model
**Ensure:** $Model_{FT}$ is fine-tuned model dictionary per group

**Step 1: Generate Synthetic Data**
**for** each group $\mathcal{G}, \mathcal{G}'$ **do**
    $T_{aug,\mathcal{G}}$ = []
**for** each conversation $C_i \in$ **C** **do**
    $\mathcal{I}_i = \texttt{Intent}(C_i)$
    $\mathcal{G}_i = \texttt{Group}(C_i)$
    $T_{aug,\mathcal{G}_i} = T_{aug,\mathcal{G}_i} \cup \{C_i\}$
    **for** $j = [1..t_i]$ **do**
        $S_j = [U_1, A_1, \ldots, U_j]_{C_i}$
        $S_{j,aug}$ = RubricGuidedDataGeneration($S_j$, $\mathcal{I}_i$,$\mathcal{G}_i$, $\mathcal{R}[\mathcal{I}_i]$)
        $T_{aug,\mathcal{G}_i} = T_{aug,\mathcal{G}_i} \cup \{S_{j,aug}\}$
**Step 2: FineTune LLM for each group**
$Model_{FT}$ = { }
**for** each group $\mathcal{G}, \mathcal{G}'$ **do**
    $Model_{FT}[\mathcal{G}_i]$ = $\texttt{FineTuneLlm}(Model_{Base}, T_{aug,\mathcal{G}_i}, \mathcal{J})$

**Algorithm 4** Rubric-Guided Data Generation

**Require:** Training example $T = [S_i, A_j, \mathcal{J}(S_i)]$, where $S_i = [U_1, A_1, \ldots, U_j]$ is a conversation up to $j^{th}$ user utterance, $A_j$ is the AI response, and $\mathcal{J}(S_i)$ is the user's preference judgement for $A_j$
**Require:** Intent $\mathcal{I}_i$, Group $\mathcal{G}_i$, Rubric $\mathcal{R}_{\mathcal{I}_i}$
**Ensure:** Augmented training data $T_{aug}$

# Generate Augmented Training Example with Rubric
**if** $\mathcal{J}(S_i)$ == +1 **then**
    # Output is preferred by user, modify to include dispreferred group aspects
    $A_{aug}$ = $\texttt{LLM.ModifyPrompt}(S_i, \mathcal{G}', \mathcal{R}_i)$
    $T_{aug} = [S_i, A_{aug}, -1]$
**if** $\mathcal{J}(S_i)$ == -1 **then**
    # Output is dispreferred by user, modify to include preferred group aspects
    $A_{aug}$ = $\texttt{LLM.ModifyPrompt}(S_i, \mathcal{G}_i, \mathcal{R}_i)$
    $T_{aug} = [S_i, A_{aug}, +1]$
**return** $T_{aug}$

**Algorithm 5** GPA-FT: Inference

**Require:** Partial conversation $S_i = [U_1, A_1, \ldots, U_j]$ up to $j^{th}$ user utterance
**Require:** Per-group, fine-tuned model dictionary $Model_{FT}$
**Ensure:** LLM answer $A_j$

**Step 1: Classify user group**
$\mathcal{G}_i = \texttt{Group}(S_i)$
**Step 2: Retrieve Group-Aware Model and generate response**
$Model_{FT}$ = $Model_{FT}[\mathcal{G}_i]$
$A_j$ = $Model_{FT,\mathcal{D}_i}(S_i)$
**return** $A_j$

clusters of intents formed within mini-batch $B_k$: $\mathcal{C}_k = \text{Cluster}(\{I_{k1}, I_{k2}, \ldots, I_{kl}\})$. After processing each mini-batch $B_k$, the intent clusters $\mathcal{C}_k$ are updated to refine clustering based on the new batch of data. The update can be represented as: $\mathcal{C}_{k+1} = \text{Update}(\mathcal{C}_k)$. This process continues iteratively as each mini-batch is processed. Once all mini-batches have been processed, the final set of intent clusters, $\mathcal{C}_{\text{final}}$, is obtained. This set represents the most prominent clusters of intents after all conversations have been analyzed. Here, $\mathcal{C}_{\text{final}}$ is the final set of intent clusters that are most significant across the entire conversation history. This mathematical framework outlines the process of extracting, clustering, and refining intents from a pool of conversations. By dividing the data into mini-batches and iteratively updating the clusters, we can identify the most prominent intents that reflect the dominant themes within the conversation history.

## D Prompts

The GPA-CT and GPA-FT methods use a number of LLM prompts which we describe in this appendix. User expectation is inferred for both satisfaction and dissatisfaction. The prompt for inferring user expectation for satisfaction in line 8 of Algorithm 1 can be found in the prompt in Figure 11 titled *LLM.InferUserExpectation (During SAT)*. Similarly, inferring the user expection in line 10 of Algorithm 1 under the dissatisfaction condition can be accomplished using the following prompt the prompt titled *LLM.InferUserExpectation (During DSAT)*. The Likert Rating is computed using the *LLM.ComputeLikertRating* prompt in Figure 13. The *ModifyPrompt* in Figure 14 is used to generate a response from the LLM. This prompt is used for GPA-CT inference in Algorithm 2 and *Rubric-Guided Data Generation* in Algorithm 4. The *Persona* results are generated with the *PersonaEvaluation* prompt presented in Table 17 and Table 18. The win-lose-tie rates for the results presented in Section 5 is shown in the prompt titled *PersonaEvaluationWinRate* in Figure 18. Figure 16 provides the *Individual DSAT Evaluation Prompt*. Finally, labelling a user's expertise in a conversation is accomplished using the *LLM.ExpertiseLabellingPrompt* prompt in Figure 15.

## E Expert and Novice Preferences

Preference examples for experts and novices for the Education and Programming domains can be found in Figure 6.

## F GPA Results for Gemma

Many of the results presented in Section omputed by finetuning LLaMA. In Table 19, we report GPA-CT and GPA-FT results for the Gemma model. The results demonstrate that both the GPA-CT and GPA-FT methods perform well when Gemma is used as the base model.

## G Generated Conversational Examples

To provide additional insights to the reader, we report example generated conversations from Education Domain for the Novice group. Conversational examples from the Base, Persona, Static, and GPA-CT models are presented, in addition to the original conversation. Furthermore, we provide the reasons and scores returned by the GPT4o judge.

### Original

**User:** Hi Bing! What is shared-living?
**Agent:** Hello, this is Bing. Shared-living is a term that describes living with others in a way that benefits one's health, social life, and environment. Some examples of shared-living are **cohousing communities**, where people share common spaces and facilities, and **student houses**, where students live together and share expenses. According to some sources[1][2][3], shared-living can offer advantages such as lower costs, more social support, and better preservation of historic buildings.
**User:** Thanks Bing
**Agent:** You're welcome. Is there anything else I can help you with?

**User:** What is the meaning of major and minor in the context of American education?
**Agent:** In the context of American education, a **major** is the main field of study that a student chooses to pursue during their bachelor's degree, while a **minor** is a secondary or complementary field of study that requires fewer courses[1][4]. For example, a student can major in engineering and minor in economics. The terms major and minor reflect the increasing specialization and diversity of the undergraduate curriculum in American higher education[5]. (Labelled as SAT)

**User:** Thanks Bing!
**Agent:** You're welcome. Have a great day!

### Base

You're welcome! In American education, **major** and **minor** refer to areas of study within a college or university degree program.

- **Major:** This is your primary field of study. You'll take the majority of your courses in this subject, and it's what you'll officially graduate with a degree in (e.g., a Bachelor of Arts in History).

- **Minor:** This is a secondary area of study that you choose to complement your major. You'll take fewer courses in a minor compared to your major, but it allows you to explore another subject of interest and gain a broader skill set (e.g., a History major with a minor in Political Science).

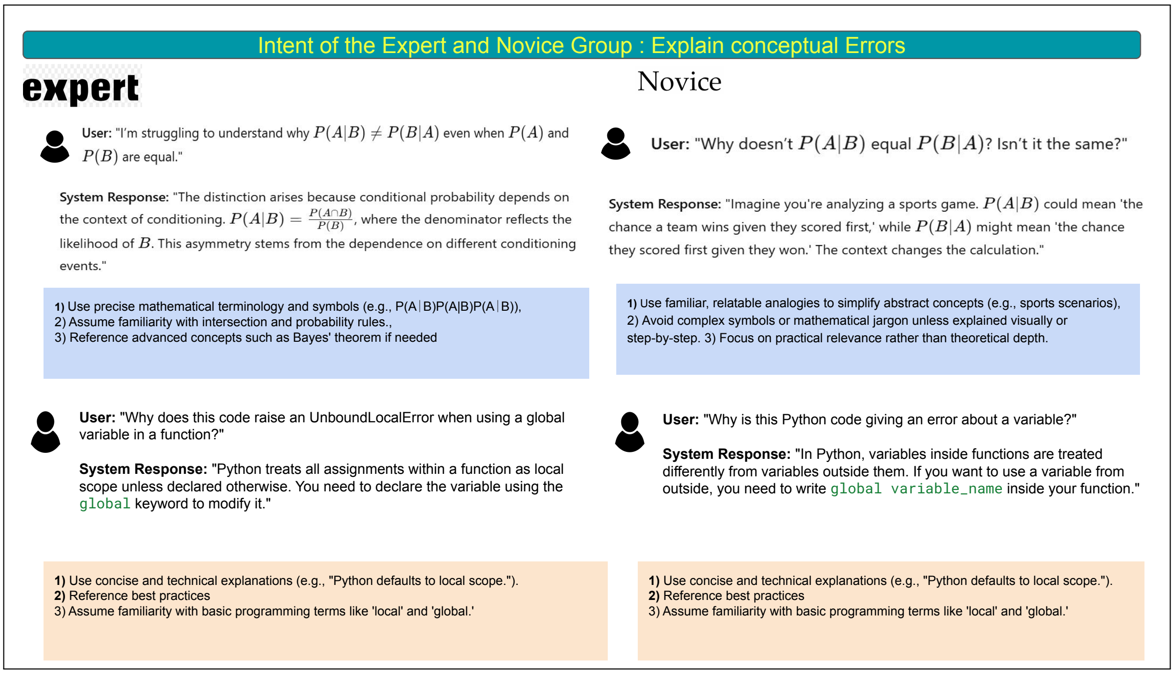


Figure 6: Comparison of expert and novice preferences across the *education and programming domains* for the intent of explaining conceptual errors. The extracted rubrics dynamically adapt based on expertise level and domain, demonstrating the effectiveness of preference extraction in context-specific learning scenarios, when novices might have similar expectations but expectations of experts vary.

Choosing a major and minor can help you focus your academic path, explore your interests, and prepare for your future career. Let me know if you have any other questions!

### Persona

Hey there! "Major" and "minor" in American education refer to the subjects you focus on during your college years. Think of it like this:

- **Major:** This is your main area of study. It's what you'll spend most of your time learning about and what will ultimately be listed on your diploma. For example, if you major in Computer Science, you'll take lots of classes about programming, algorithms, and software development.
- **Minor:** This is a secondary area of study that you choose in addition to your major. It's a way to explore another subject you're interested in and gain some extra knowledge. So, if you major in Computer Science, you could minor in something like Mathematics or Art History.

Choosing a major and minor helps you specialize in a field you're passionate about and build a well-rounded education. Hope that clears things up! Let me know if you have any other questions.

### Static

Hey there! "Major" and "minor" are like specializations you choose when you're in college. Think of it like this: you're building a house.

- **Major:** This is the main part of the house, like the bedrooms, living room, and kitchen. It's the biggest part and what you'll spend most of your time studying. It's what you want to become an expert in, like computer science, history, or biology.
- **Minor:** This is like a smaller room, like a study or a sunroom. It's something you're interested in, but not as much as your major. It could be something related to your major, like math for a computer science major, or something completely different, like music.

So, you pick a major to focus on, and you can choose a minor to explore something else you like!

### CT

In American education, "major" and "minor" describe your areas of focus during college. Think of

your **major** as your primary area of study. You'll take the majority of your classes in this subject and earn your degree in it (like a Bachelor of Arts in English or a Bachelor of Science in Computer Science).
A **minor** is a smaller area of study that you choose to complement your major. You'll take fewer classes in this subject, but it demonstrates knowledge in a different field. It can make your resume stand out to potential employers or broaden your overall education.
For example, if you're majoring in History, you might minor in Political Science.

### CT vs Base

"Reason": "The user is a novice in the education domain, and the conversation history indicates they are seeking clear and straightforward explanations. Both options provide a definition of major and minor in the context of American education, but Option A is more concise and uses simpler language, which is more suitable for a novice. Option A also provides examples that are easy to understand. This approach aligns with the user's need for a basic understanding without overwhelming them with too much detail. Option B, while also clear, uses slightly more complex language and structure, which might be less accessible to a novice. Therefore, Option A is preferred for its simplicity and clarity.", "Output": "Option A", "Confidence": 75

### CT vs Persona

"Reason": "The user is a novice in the education domain, and the response should be clear and informative. Option A provides a straightforward explanation of what a major and minor are, with examples that illustrate the concept. It explains the primary and secondary focus of study and how they complement each other. Option B, while also informative, includes more examples and additional benefits of having a major and minor, which might be slightly more detailed than necessary for a novice. However, both options are suitable, but Option A is more concise and directly addresses the user's question without additional information that might overwhelm a novice.", "Output": "Option A", "Confidence": 75.

### CT vs Static

"Reason": "The user is a novice in the education domain, as indicated by the task instruction. Option A provides a straightforward and concise explanation of the terms 'major' and 'minor' in American education, which is suitable for someone new to the field. It clearly defines both terms and provides simple examples to illustrate the concepts. Option B, while also informative, is more detailed and includes additional information that might be overwhelming for a novice. It also uses more technical language and provides an example that might be more suited for someone with a bit more background knowledge. Given the user's novice status, Option A is more appropriate as it is easier to understand and directly addresses the user's likely level of knowledge.", "Output": "Option A", "Confidence": 75.

## H Additional Results

Both GPA-FT and GPA-CT can generate more satisfactory responses than the original DSAT responses. We compare each of the model responses with the actual DSAT responses from Experts and Novice Samples and prompt GPT4-o to decide which of the response would increase satisfaction level given the conversation history and the followup dissatisfactory user feedback. Next, we observe whether our methods are chosen over the true dissatisfactory responses and we show results in Table 21. We found that compared to baselines, GPA-FT and GPA-CT provides higher winrates.

In Figure 7, we plot how the training set size used to compute the rubric for GPA-CT affects the WinRate (WR).

**DPO with preference data is the optimal choice for GPA-FT.** Table 20 clearly shows that DPO consistently achieves the highest win rates, making it the best choice for the GPA-FT approach. DPO vs. Base achieves a 73.36% Win Rate, significantly outperforming other methods, demonstrating that DPO leads to stronger preference alignment compared to the base model. Additionally, DPO vs. KTO-Only (70.61% WR) and DPO vs. KTO-Augmented (61.46% WR) indicate that DPO still maintains a strong advantage over KTO-based approaches, reinforcing its robustness in fine-tuning. Rubric augmentation using KTO does help, as KTO-Augmented vs. Base achieves a 67.63% Win Rate, showing an improvement over KTO vs. Base (53.41%). However, KTO-based models still fall short of DPO's performance, especially when comparing DPO vs. KTO-Augmented, where DPO wins 61.46% of the time. This suggests that while

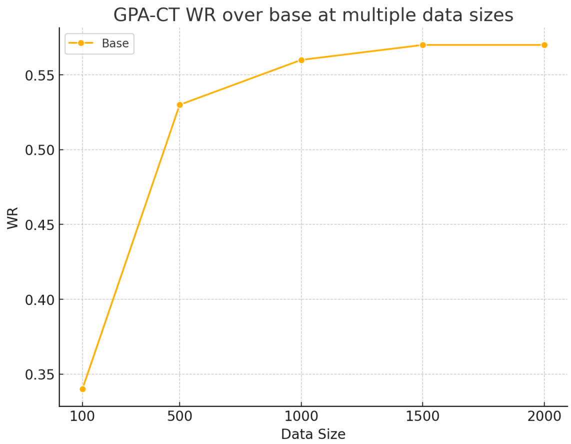


Figure 7: Learning Curve to Show that GPA-CT is a data-efficient algorithm. We vary the training data size (100, 500, 1000, 1500, 2000 samples) for extracting our preference rubrics across the Expert and Novice Groups on Wildhat Programming Domain and observe the Win-Rate over the base model on a held-out test set using Prompt 17. We observe the performance becomes stable using only 1000 examples for rubric creation, making it a data-efficient customization approach.

rubric augmentation improves alignment, it is not as effective as preference optimization through DPO. Thus, DPO is the optimal choice for GPA-FT.

## I Note on Generalizability

**GPA is generalizable across domain and user-groups.** Results in Table 7 and Table 8 provide insights into GPA-CT's generalization across different cultural and expertise-based user groups, evaluated at varying EC (confidence) thresholds (65, 70, 75). The results compare GPA-CT against Base, Persona, and Static setups in two domains: India vs. US (Cultural Evaluation) and Education (Novice vs. Expert Evaluation).

1. Cultural Generalization (India vs. US) Consistent Gains Over Base: GPA-CT consistently outperforms the Base model across EC thresholds, with a win rate increasing from 60.57% (EC=65) to 63.52% (EC=75), showing stable adaptation across cultural contexts. Stronger Performance Against Persona Static: The win rate vs. Persona remains stable (65%), while GPA-CT shows stronger gains vs. Static at EC=70 (72.22%), indicating that contextual fine-tuning provides advantages over rigid static responses. Higher EC Improves Differentiation: The lose rate decreases slightly at EC=75, suggesting that higher confidence filtering leads to better generalization. 2. Expertise Generalization (Novice vs. Expert in Education Domain) Performance Is More Balanced: Compared to cultural evaluation, win rates in the education domain are more balanced across expertise groups, suggesting that novices and experts respond similarly to GPA-CT's responses. Slight Gains Over Persona: GPA-CT vs. Persona win rate remains around 51-52%, indicating Persona-based fine-tuning is already well-aligned with education-based responses. Better Adaptation Over Static: GPA-CT outperforms Static consistently (55.19% win rate at EC=65 and 54.90% at EC=75), reinforcing the advantage of dynamic context-aware models over fixed prompts.

GPA-CT generalizes well across cultural differences (India vs. US), maintaining a consistent win rate across EC thresholds. Higher EC filtering enhances performance differentiation, particularly in cultural evaluations. Performance in the education domain is more balanced across expertise levels, with smaller gains over Persona but consistent advantages over Static. EC thresholds affect generalization differently—higher EC benefits cultural evaluations more than expertise-based evaluations.

| User Groups | Intent | Rubric Item | Description |
|---|---|---|---|
| US vs China | Writing Assistance | Personal Connection and Passion | Western users seek vivid, personal engagement, while Eastern users prefer clear and concise communication, emphasizing empathy and understanding. |
| | | Historical and Anecdotal Content | Western users favor detailed historical accounts with personal anecdotes, while Eastern users prefer straightforward summaries with clear information. |
| | | Perspective and Tone | Western users prefer second-person perspective, addressing the audience directly, while Eastern users expect the bot to acknowledge and appreciate their contributions. |
| | | Refinement in Narrative Style | Western users prefer advanced vocabulary and polished narrative styles, while Eastern users value clarity, conciseness, and brevity. |
| US vs China | Creative Content Creation | Story Continuation | US users prefer detailed and structured script outlines, while Eastern users expect more imaginative and action-packed continuations. |
| | | Role-Playing Engagement | US users may expect the assistant to ask for specifications, while Eastern users expect immediate role-play engagement. |
| | | Humour and Creative Titles | Both groups enjoy humorous and creative titles, but Eastern users emphasize playful and whimsical text more. |
| | | Cultural Resonance and Poetic Elements | Both groups value cultural resonance, but Eastern users place more emphasis on poetic elements. |
| US vs India | Writing Assistance | Acknowledgment and Appreciation | Indian users expect explicit appreciation and acknowledgment of their contributions, while US users do not emphasize this as much. |
| | | Personal Connection and Passion | US users prefer vivid engagement with emotions and enthusiasm, while Indian users prioritize shared goals and inclusive language. |
| | | Engaging and Descriptive Style | US users prefer engaging and descriptive styles with coherence, while Indian users focus on vivid and friendly tones. |
| US vs India | Creative Content Creation | Story Continuation | US users prefer structured and detailed script outlines, while Indian users expect more imaginative and action-packed stories. |
| | | Bedtime Story Personalization | US users expect generic stories, while Indian users prefer more personalized and interactive bedtime storytelling. |

Table 5: Rubric Items Differentiating the Preferences Across User Groups (Separated by Country/Cultural Context) in the domain of Creative Writing and Editing where Likert Scale rating is greater than 3.

| User Groups | Intent | Rubric Item | Description |
|---|---|---|---|
| US vs China | Information Seeking | Grammatical Explanation | Both user groups expect clear and accurate explanations of grammatical correctness. There is no observed difference in their expectations regarding this aspect |
| | | Clarification of Translations | Both groups expect clear and accurate clarifications regarding translations of book titles. There is no observed difference in their expectations for this aspect." |
| | | List of Frequently Quoted Sentences | Both groups expect well-organized and relevant lists of frequently quoted sentences from books. There is no observed difference in their expectations for this aspect. |
| | | Explanation of Setting and Plot | Both groups expect clear explanations of how the setting enhances the plot, but Western users may prefer more straightforward and concise answers. This results in a minor difference in expectations. |

Table 6: Rubric Items Differentiating the Preferences Across User Groups (Separated by Country/Cultural Context) in the domain of Creative Writing and Editing where Likert Scale rating is less than 3, signifying no major difference in expectations/preferences when their intent is to seek information in the domain of Creative Writing and Assistance.

| Model | EC = 65 | EC = 70 | EC = 75 |
|---|---|---|---|
| **Llama-India and US** | | | |
| GPA-CT vs Base | 0.6057 / 0.3942 | 0.6038 / 0.3961 | 0.6352 / 0.3647 |
| GPA-CT vs Persona | 0.6490 / 0.3509 | 0.6473 / 0.3526 | 0.6666 / 0.3333 |
| GPA-CT vs Static | 0.6473 / 0.3526 | 0.7222 / 0.2661 | 0.6666 / 0.3333 |

Table 7: Performance comparison of GPA-CT models prompt-tuned with India and US Persona at different EC levels

| Model | EC = 65 | EC = 70 | EC = 75 |
|---|---|---|---|
| **Education Domain** | | | |
| GPA-CT vs Base | 0.5259 / 0.4740 | 0.5259 / 0.4740 | 0.5229 / 0.4771 |
| GPA-CT vs Persona | 0.5140 / 0.5259 | 0.5140 / 0.5259 | 0.5271 / 0.5229 |
| GPA-CT vs Static | 0.5519 / 0.4481 | 0.5519 / 0.4481 | 0.5490 / 0.4599 |

Table 8: Performance comparison of models in the Education Domain at different EC thresholds.

| **LLM.StaticPrompting** |
|---|
| **[1]** Overview<br>Tell me the expectations of a persona in a domain from the chatbot. Answer in a few sentences. |

Table 9: Static

| **LLM.PersonaRole-Playing** |
|---|
| **[1]** # OVERVIEW<br>You will be given a conversation between a User and an AI agent. Your task is to generate response that would tailor to a persona in the domain domain. |

Table 10: Persona-Role Playing Response Generation

| LLM.InferUserExpectation (During SAT) |
|---|
| **[1]** # OVERVIEW<br>You will be given a conversation between a User and an AI agent. Your task is to assess the reasons of user's happiness based on the conversation history and the bot response.<br># TASK:<br>Classify the user's intent from the conversation conversation history. Also determine what the user expects from the bot and why the user finds the bot's response user remarks useful. Determine based on whatever the user remarks after the bot's response user remarks.<br># ANSWER FORMAT<br>Format your output as JSON Object where the keys are user-intent, user-expectation-from-bot and reasons-for-happiness. Do not output anything else except this. |

Table 11: LLM.InferUserExpectation (During SAT)

| LLM.InferUserExpectation (During DSAT) |
|---|
| **[1]** # OVERVIEW<br>You will be given a conversation between a User and an AI agent. Your task is to assess the reasons of user's frustration based on the conversation history and the bot response.<br># TASK:<br>Classify the user's intent from the conversation conversation history. Also determine what the user expects from the bot and why the user finds the bot's response user remarks frustrating. Determine based on whatever the user remarks after the bot's response user remarks.<br># ANSWER FORMAT<br>Format your output as JSON Object where the keys are user-intent, user-expectation-from-bot and reasons-for-frustration. Do not output anything else except this. |

Table 12: LLM.InferUserExpectation (During DSAT)

| **LLM.ComputeLikertRating** |
|---|
| **[1]** # OVERVIEW<br># Task Overview:<br>You have to compare expectations of two user groups based on some aspects in {domain-name}, and provide ratings of 1-5 depending on how much different are their expectations from the bot while interaction. You have to update the comparison output based on what was observed previously {previous-observations} and the current observed differences in expectations between group 1 and group 2 described below. Make sure that if there is no observed datapoints for an aspect in either expert or novice category, provide the least rating in that case.<br># Primary Intent<br>Intent : {intent-cluster-of-the-user}<br># Expectations of Group 1<br>{expectation-of-group 1}<br># Expectations of Group 2<br>{expectation-of-group 2}<br># Annotation Guidelines on a scale of 1-5<br>1 : It indicates there is no observed difference between the expectation of two groups on this aspect,<br>2 : It indicates there is a minor difference between the expectation of two groups on this aspect,<br>3 : It indicates moderate difference between the expectation of two groups on this aspect,<br>4 : It indicates remarkable difference between the expectation of two groups on this aspect,<br>5 : It indicates undoubetedly stark difference between the expectation of two groups on this aspect.<br># Output Format<br>Format your output as JSON where keys are aspects and values are 1) ratings from 1-5 and 2) Interpretation of the rating in 2-3 sentences. |

Table 13: LLM.ComputeLikertRating

| **LLM.ModifyPrompt** |
|---|
| **[1]** # OVERVIEW<br># Task<br>You will be provided with a conversation between a user and bot. Based on the conversation history, you have to generate a suitable response. Make sure that you follow some rules while generating the response.<br># Conversation History<br>{conversation-history}<br># User Input<br>{user-input}<br># Some Rules to Follow<br>{reminder}<br># Output Format<br>Format your output as a JSON Object with response as key. Do not output anything else except this JSON. |

Table 14: LLM.ModifyPrompt

| **LLM.ExpertiseLabelling** |
|---|
| **[1]** # OVERVIEW<br># OVERVIEW<br>You will be given a conversation history between a User and an AI agent. Your task is to determine user's expertise in the subject of the conversation.<br># USER EXPERTISE<br>User expertise levels in a conversation subject range from novice, indicating a lack of familiarity with fundamental concepts, to expert or master, denoting a deep understanding of relevant vocabulary, concepts, and principles.<br>- Novice: A subject novice is a person who has little or no familiarity with a specific topic or domain. A subject novice may ask questions that are vague, general, irrelevant, or based on incorrect assumptions. A subject novice may also have difficulty understanding the terminology, concepts, or arguments of experts or more knowledgeable people in the subject. They may ask basic or general questions that can be answered by simple definitions, examples, or facts. They may not be aware of the sources, methods, concepts, or terminology that are relevant to the subject.<br>- Intermediate: A subject intermediate is someone who has some basic knowledge or familiarity with a certain topic, but not enough to be considered an expert or a novice. A subject intermediate can ask general questions that reflect their curiosity or interest in the topic, but not very specific or complex ones that require deeper understanding or analysis. A subject intermediate might have learned some terms or concepts related to the topic, but not how to apply them in different contexts or situations.<br>- Expert: A subject expert is someone who can apply relevant concepts and terminology to different scenarios and problems. They can analyze and interpret data, compare and contrast different methods or approaches, and justify their reasoning with evidence. The user also demonstrates curiosity and interest in the subject by asking questions that go beyond the surface level and explore the deeper implications and connections of the topic. He has a deep and comprehensive understanding of a specific topic or field, and can use specialized terms and references to communicate their knowledge. A subject expert can state accurate facts, provide relevant examples, and cite authoritative sources related to their topic or field.<br>- Unknown: There is not enough information to determine the user's expertise.<br>## OUTPUT FORMAT ##<br>Format your output as JSON Object with key as Expertise-label and values as either Novice, Intermediate, Expert or Unknown.<br>## INPUT ##<br>Conversation History<br>## OUTPUT ## |

Table 15: Prompt to Classify Expertise Labels

| **LLM.IndividualDSATEvaluation** |
|---|
| **[1]** # OVERVIEW<br># Task<br>In the conversation context between user and assistant: {conversation_history}, based on user utterance : {user_utterance}, when the bot responds : {bot_response}, the user felt {label}, then he provides a feedback by commenting {feedback_comment}.<br>You have to compare Option A and Option B and judge which response is very different from reference bot response {bot_response}, such that the user will not provide a followup comment {feedback_comment}.<br># Option A<br>{optionA}<br># Option B<br>{optionB}<br># Output Format<br>Format your output as a JSON Object with keys as Option and reasoning. Output either Option A or Option B or can't decide. You should not output anything except the JSON. Do not judge based on user expertise. Judge only based on which response is very different from reference bot response {bot_response}. |

Table 16: Individual DSAT Evaluation

| **LLM.PersonaEvaluationwithEC** |
|---|
| **[1]** OVERVIEW<br>Task<br>Imagine yourself as a persona in the domain domain. Based on your persona and the conversation history, you have to judge which response would you prefer among Option A and Option B along with the step-by-step reasoning.<br>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:<br>1–20 (Uncertain): The user profile provides insufficient or minimal evidence. The decision is largely based on weak or indirect hints.<br>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.<br>61–80 (Confident): The user profile contains strong evidence that clearly supports your prediction, with very little ambiguity.<br>81–100 (Highly Confident): The user profile provides direct and explicit evidence that decisively supports your prediction<br># Conversation History<br>{conversation$_{history}$}<br># Option A<br>option1<br># Option B<br>option2<br># Output Format<br>Format your output as a JSON Object with keys as Reason, Output, Confidence. Output the step-by-step reasoning and then Option A or Option B and the confidence value from 1-100. You should not output anything except the JSON. |

Table 17: Prompt used for LLM-as-a-Personalized-Judge as borrowed from (Dong et al., 2024)

| **LLM.PersonaEvaluationwithoutEC** |
|---|
| **[1]** OVERVIEW<br>Task<br>Imagine yourself as a persona in the domain domain. Based on your persona and the conversation history, you have to judge which response would you prefer among Option A and Option B along with the step-by-step reasoning.<br>Task<br>Imagine yourself as a persona in the domain domain. Based on your persona and the conversation history, you have to judge which response would you prefer among Option A and Option B along with the step-by-step reasoning.<br># Conversation History<br>{conversation-history}<br># Option A<br>option1<br># Option B<br>option2<br># Output Format<br>Format your output as a JSON Object with key as Reason and Output. Output the step-by-step reasoning and then Option A or Option B or can't decide. You should not output anything except the JSON. |

Table 18: Prompt used for LLM-as-a-Personalized-Judge with tie

| **Model** | **LLM Pref (W/L/T)** | **LLM conf $\geq$ 75** | **LLM Pref (W/L/T)** | **LLM conf $\geq$ 75** |
|---|---|---|---|---|
| | **Intent=Programming/Group=Novice** | | **Intent=Programming/Group=Expert** | |
| GPA-CT vs Base | 0.6930 / 0.3041 / 0.0029 | 0.5797 / 0.4203 | 0.4489 / 0.5426 / 0.085 | 0.4696 / 0.5304 |
| GPA-CT vs Persona | 0.5702 / 0.4298 / 0.0000 | 0.5430 / 0.4570 | 0.5824 / 0.4063 / 0.114 | 0.5924 / 0.4076 |
| GPA-CT vs Static | 0.5754 / 0.4187 / 0.0058 | 0.5300 / 0.4700 | 0.5625 / 0.4290 / 0.085 | 0.6364 / 0.3636 |
| GPA-FT vs Base | 0.6842 / 0.2953 / 0.0205 | 0.6426 / 0.3574 | 0.5966 / 0.4034/ 0.0000 | 0.5676 / 0.4324 |
| GPA-FT vs Persona | 0.6439 / 0.4561/ 0.0000 | 0.5290 / 0.4710 | 0.6676 / 0.3295 / 0.0028 | 0.6939 / 0.3061 |
| GPA-FT vs Static | 0.6433 / 0.3480 / 0.0088 | 0.5807 / 0.4193 | 0.6761 / 0.3210 / 0.0028 | 0.6842 / 0.3158 |
| GPA-FT vs GPA-CT | 0.5351 / 0.4649 / 0.0000 | 0.5426 / 0.4574 | 0.6903 / 0.3097 / 0.0000 | 0.6764 / 0.3236 |

Table 19: GPA-CT and GPA-FT results on the BingChat dataset for Gemma for the Novice and Expert groups. The LLM expected confidence $\geq$ 75 is reported and W/L/T=win/lose/tie.

| Comparison | Win % | Lose % |
|---|---|---|
| DPO vs Base | 73.36% | 26.64% |
| DPO vs KTO-Augmented | 61.46% | 38.54% |
| DPO vs KTO-Only | 70.61% | 29.39% |
| KTO vs Base | 53.41% | 46.59% |
| KTO-Augmented vs Base | 67.63% | 32.37% |

Table 20: Win/Loss Percentages of Different Finetuning Methods on BingChat Test Set justifying our best choice of DPO for our remaining GPA-FT experiments.

| | Win Rates | Lose Rates | Tie Rates |
|---|---|---|---|
| `GPA-FT` | 0.4848 | 0.2504 | 0.2648 |
| `GPA-CT` | 0.4877 | 0.2470 | 0.2653 |
| Static | 0.4283 | 0.3014 | 0.2703 |
| Persona | 0.4252 | 0.2491 | 0.3257 |
| Base | 0.4333 | 0.5167 | 0.0500 |

Table 21: Normalized Win, Lose, and Tie Rates