# GSSPO: Group Sentence-Specified Sequence Policy Optimization for LLM Personality Alignment

**Shurui Zhang***
The Hong Kong University of Science and Technology
szhangfa@connect.ust.hk

**Zikai Li***
The Chinese University of Hong Kong
zikaili@link.cuhk.edu.hk

## Abstract

Large Language Models (LLMs) have demonstrated strong reasoning capabilities, yet achieving nuanced personality alignment for simulating human conversation remains challenging. This difficulty stems from two bottlenecks: the scarcity of scalable, privacy-respecting training data and the limited ability of conventional reinforcement learning (RL) rewards to capture subtle personality-relevant signals. In this work, we introduce a unified framework to address these issues. First, we propose the Personality Alignment Choice Forge (PACF), an automatic pipeline that transforms existing conversational datasets into structured, multiple-choice benchmarks, constraining the model's action space and enabling scalable evaluation grounded in observed human choices. Second, we present Group Sentence-Specified Sequence Policy Optimization (GSSPO), an RL algorithm designed to address the coarse credit assignment problem in methods that apply a uniform reward across all tokens. By integrating stable sequence-level updates with differentiated, rubric-guided token-level feedback, GSSPO assigns reward signals to more localized spans of the response. This targeted approach mitigates optimization instabilities caused by undifferentiated reward signals, leading to more effective training in our experiments. On the Qwen3 model family, GSSPO achieves the best personality-alignment accuracy on every benchmark and scale in the SFT+RL regime, and leads the other RL methods on five of six in the SFT-Free regime. These results suggest a scalable route toward training LLMs that better match measured personality-alignment preferences. Our core code is available at https://github.com/HKUST-FintechLab/GSSPO

## 1 Introduction

Large Language Models (LLMs) have demonstrated substantial potential over a wide spectrum of complex reasoning tasks, with strong performance in mathematical reasoning (Lewkowycz et al., 2022) and chain-of-thought style multi-step problem solving (Wei et al., 2023), and further gains from scaling and improved training recipes in frontier models (OpenAI, 2024). They can also be combined with external tools to solve harder compositional problems—such as API and tool use (Schick et al., 2023), and they improve on multi-step problems under zero-shot chain-of-thought prompting (Kojima et al., 2023). However, advancing the general reasoning capabilities of LLMs remains an ongoing challenge (Zhao et al., 2025b; Huang and Chang, 2023). To address this, recent Reinforcement Learning with Verifiable Rewards (RLVR) methods show that verifiable rewards can improve reasoning-oriented behavior in LLMs (Lambert et al., 2025; DeepSeek-AI et al., 2025; Team et al., 2025).

Simulating real human conversation with LLMs is increasingly important for building natural and credible interactive systems, especially where conversation quality shapes trust, safety, and user outcomes. Typical deployments include LLM "user simulators" for training and evaluating task-oriented dialogue agents in customer support, and role-play patients or students for practicing clinical interviewing and tutoring strategies (Almansoori et al., 2025; Sekulić et al., 2024). However, LLMs remain limited in tasks that require consistent simulation of individual conversational preferences, motivating further study of **LLM personality alignment**. This field still faces numerous challenges (Larooij and Törnberg, 2025; Zhao et al., 2025a). Among these, training LLMs to better match individual-level conversational preferences remains a central challenge, and reasoning-oriented techniques do not necessarily improve LLMs' role-playing ability (Feng et al., 2025).

Two challenges remain central. First, scalable personality-alignment data are difficult to obtain:

* Equal contribution.



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 16331–16362
October 24-29, 2026 ©2026 Association for Computational Linguistics

collecting real profiles and preference signals is costly, legally constrained, and privacy-sensitive. Although recent resources collect user profiles, conversations, or personalized feedback (Kirk et al., 2024; Zhu et al., 2025; Zhao et al., 2025c), they are not always organized for large-scale training and automatic evaluation.

Second, existing RL objectives (Schulman et al., 2017; Shao et al., 2024; Zheng et al., 2025) often rely on coarse sequence-level rewards. Personality signals are subtle and context-dependent, so a single scalar score may not identify which spans support or undermine the final choice. This motivates reward assignment methods that remain scalable while providing more localized supervision for personality-relevant behavior.

Our work presents a unified framework for training language models to better match specified personality-conditioned preferences while addressing action-space size and imprecise credit assignment. Our proposed methodology, comprising the **Personality Alignment Choice Forge (PACF)** and **Group Sentence-Specified Sequence Policy Optimization (GSSPO)**, operates along two complementary axes. (1) **Choice-structured alignment formulation:** PACF reformulates open-ended conversational alignment as a discrete, multiple-choice decision problem grounded in observed human selections. By generating calibrated choice sets with high-quality distractors, this stage constrains the model's effective action space and enables discrete reinforcement supervision. (2) **Granularity-aware reinforcement optimization:** Building on this foundation, GSSPO introduces a hybrid reinforcement learning objective that combines stable, sequence-level updates with criterion-aware, token-level feedback. By employing structured, rubric-based judgments from a learned evaluator, GSSPO adaptively distributes rewards across tokens, enabling more targeted correction of personality-relevant behaviors while maintaining training stability in our experiments. Our main contributions are summarized as follows:

(i)**PACF(Personality Alignment Choice Forge): an automatic dataset generation pipeline.** We introduce a **choice-based formulation for personality alignment**, providing a general, dataset-agnostic pipeline that transforms human conversational preferences into automatically evaluable benchmarks supporting both in-domain and out-of-domain persona evaluation.

(ii)**GSSPO: a granularity-aware optimization technique.** We propose GSSPO, a reinforcement learning method that combines sequence-level optimization with localized token-level corrections for LLM training. Compared with prior uniform reward assignment methods, GSSPO unifies sequence-level and fine-grained token-level optimization through rubric-guided rewards, enabling more controllable and interpretable alignment learning.

## 2 Related Work

**Personality Alignment.** Personality alignment has been studied through datasets and optimization methods that connect user attributes to conversational preferences. **PRISM** (Kirk et al., 2024) links participant profiles to dialogue-time feedback across a diverse cohort, showing that alignment depends on the feedback provider; Zhu et al. (2025) introduce PAPI to evaluate whether LLMs can reproduce personality-conditioned behavioral patterns. Methodologically, Zhao et al. (2025c) propose **Reinforcement Learning for Personalized Alignment (RLPA)**, in which models adapt to user profiles through interactive simulations. More broadly, preference-optimization methods such as **Direct Preference Optimization (DPO)** (Rafailov et al., 2023) and **Simple Preference Optimization (SimPO)** (Meng et al., 2024) provide offline objectives for comparative preferences: DPO optimizes preference likelihood without an explicit reward model, while SimPO uses a reference-free, length-normalized reward with a target margin. These methods motivate optimization over personality-aware preference data.

**Synthetic Data Generation.** Synthetic data generation for NLP increasingly uses instruction-tuned LLMs to produce training signals. **Self-Instruct** (Wang et al., 2023) bootstraps instruction-response corpora from small seed sets, and **Stanford Alpaca** (Taori et al., 2023) uses teacher-generated data to fine-tune smaller instruction-following models. Alignment work further extends synthetic supervision to critiques and preferences: **UltraFeedback** (Cui et al., 2024) scales preference signals, Direct Preference Optimization (DPO) (Rafailov et al., 2023) learns directly from preference pairs, and **LLM-as-a-judge** methods (Zheng et al., 2023) support automated filtering, ranking, and refinement. Together, these works establish a paradigm in which generated instructions, preference signals, and judgments supplement human annotation for

model alignment.

**Reinforcement Learning Algorithms for LLMs.** Policy-gradient methods underpin RLHF; PPO (Schulman et al., 2017) is widely used but requires careful tuning for large transformers. Group-based optimization adapts RL to text: GRPO (Shao et al., 2024) samples response groups and computes relative advantages through within-group reward normalization, removing the value model. However, token-level importance weights can become high-variance on long sequences and large models (Zheng et al., 2025). GSPO (Zheng et al., 2025) addresses this by using sequence-level ratios with length-aware PPO clipping, aligning reward assignment and policy updates at the sequence level. Complementary work studies denser supervision and richer feedback: process supervision improves over outcome-only supervision for mathematical reasoning (Lightman et al., 2023); TLCR (Yoon et al., 2024) derives token-level continuous rewards from discriminator confidence; RuscaRL (Zhou et al., 2026) uses rubric-guided rollouts with verifiable rewards; and SDPO (Hubotter et al., 2026) converts rich feedback into dense policy updates via self-distillation. These methods motivate our focus on reward granularity and stable credit assignment for personality-alignment training.

## 3 Methodology

### 3.1 PACF: Personality Alignment Choice Forge

**Motivation and formulation.** To reduce the action space of open-ended personality simulation, we propose **PACF** (Figure 4), a dataset-agnostic pipeline that converts conversational datasets into automatically evaluable *choice-based alignment benchmarks*. Given multi-turn conversations with user profiles and observed human preferences, such as Kirk et al. (2024) and Li et al. (2025), PACF reformats each preference-bearing interaction into a multiple-choice item where one option corresponds to the observed human-selected response and the remaining options are calibrated distractors. This formulation preserves grounding in human decisions while making RL training and automatic evaluation tractable.

Formally, PACF takes a conversational dataset $\mathcal{D}$ with multi-turn dialogues and associated user profiles (e.g., demographics, personality traits, or stated values), and returns a multiple-choice dataset $\mathcal{D}^{\mathrm{MC}}$. Each item is defined as

$$x = (\text{profile},\ \text{context},\ \mathcal{O} = \{o_1, \ldots, o_n\},\ y),$$

where $y$ is the index of the human-preferred option. PACF has three stages: **(A)** conversation preprocessing and structural compression, **(B)** candidate response generation and distractor-bank construction, and **(C)** multiple-choice packaging with in-domain and held-out-persona split construction. We instantiate PACF on PRISM and Big5Chat, with detailed construction procedures, distractor taxonomy, and dataset statistics provided in Appendix H.

### 3.2 GSSPO: Group Sentence-Specified Sequence Policy Optimization

To train LLMs to align with individual human personalities, we propose **Group Sentence-Specified Sequence Policy Optimization (GSSPO)** (Figure 1), a reinforcement learning algorithm that provides differentiated rewards for localized token spans. GSSPO operates in an online RL fashion similar to PPO/GRPO/GSPO: the model (policy $\pi_\theta$) is initialized (e.g. from a pre-trained base model) and iteratively updated through episodes of sampling and feedback. However, current reinforcement learning algorithms generally do not provide localized token-level feedback about which parts of a response support or undermine the final answer. Our core innovation lies in using a judgment model to evaluate whether generated responses meet specific quality criteria, and then applying different optimization granularities accordingly.

**Modeling.** Let "profile" tensor be $p$, "conversation history" tensor be $ch$, then our training aim to optimize the mapping $f_{p,ch} : S \longrightarrow O$, where $S$ is the set of sentences and $O$ is the set of options.

Here we define a criterion-based reward function $g_{cri} : E \longrightarrow \mathbb{R}$, where $E$ is the set of explanations given by LLMs. The corresponding optimization target is

$$\text{maximize } \mathbb{E}[g_{cri}(x)] \quad \text{subject to } x \in E$$

**Unified GSSPO Objective.** In sequence-level, we use optimization similar to GSPO (Zheng et al., 2025):

$$L_i(\theta) = -\min\left(s_i(\theta)\hat{A}_i, \text{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right)$$

where the sequence-level importance ratio is:

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{old}}(y_i|x)}\right)^{1/|y_i|}$$

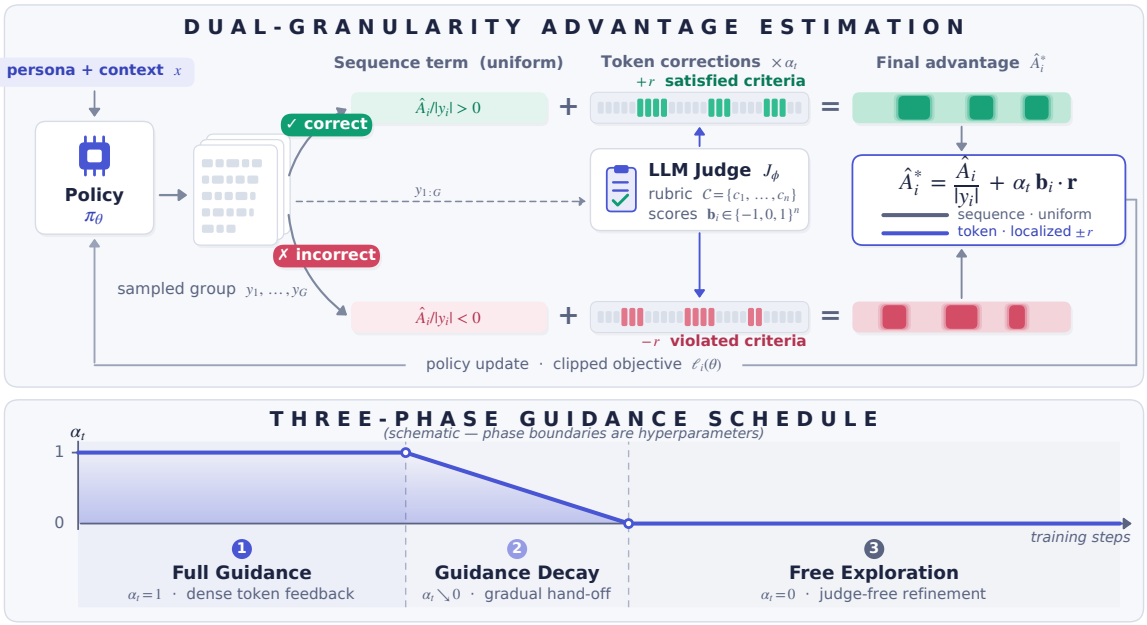



Figure 1: **Overview of the Group Sentence-Specified Sequence Policy Optimization (GSSPO) framework.** The pipeline consists of two synergistic components designed to align LLMs with multi-dimensional personality criteria. **Top: Dual-Granularity Advantage Estimation.** The model $\pi_\theta$ samples a group of responses $y_{1:G}$ which are evaluated by an external LLM judge against a criteria set $\mathcal{C}$. Unlike standard RL which assigns a uniform score to the entire sequence, GSSPO synthesizes a hybrid advantage $\hat{A}_i^*$ by combining a uniform sequence-level base term $\hat{A}_i/|y_i|$ with fine-grained *token-level corrections* $\alpha_t\,\mathbf{b}_i \cdot \mathbf{r}$. Using a hierarchical localization strategy (exact/fuzzy matching), the judge's criterion scores $\mathbf{b}_i$ are mapped to specific token spans, allowing the model to receive precise credit: $+r$ on satisfied criteria of correct responses, $-r$ on violated criteria of incorrect ones. **Bottom: Three-Phase Guidance Schedule.** The integration of token-level signals is controlled by a dynamic coefficient $\alpha_t$. The training evolves through three distinct phases: (1) *Full Guidance* for rapid rule acquisition, (2) *Guidance Decay* to progressively reduce dependency on the external judge, and (3) *Free Exploration* to refine general reasoning capabilities without token-guided constraints. The schedule is drawn schematically; phase boundaries are hyperparameters (Appendix A).

Detailed preliminary is shown in Appendix J.

The complete GSSPO optimization objective combines both granularities:

$$\mathcal{L}_{\text{GSSPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y_{1:G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\ell_i(\theta; x, y_i)\right],$$

where:

$$\ell_i(\theta; x, y_i) = -\min\left(s_i(\theta)\hat{A}_i^*, \text{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i^*\right),$$

Here the advantage $\hat{A}_i^*$ is distributed as:

$$\hat{A}_i^* = \frac{\hat{A}_i}{|y_i|} + \alpha_t \mathbf{b}_i \cdot \mathbf{r}$$

where $\alpha_t$ is the guidance coefficient at training step $t$ and controls the influence of the amending term $\mathbf{b}_i \cdot \mathbf{r}$. Here $\mathbf{b}_i \cdot \mathbf{r}$ is the criteria amending part and will be introduced in the following paragraphs.

**Group-normalized advantage and the $\alpha$-blend.** Both terms of $\hat{A}_i^*$ are produced by one pair of blended rewards. Write $R_g^{\text{resp}}$ and $R_g^{\text{token}}$ for the guided rewards returned by $J_\phi$, and $R_d^{\text{resp}}$ and $R_d^{\text{token}}$ for the defaults used when the judge is not queried. At training step $t$ the guidance coefficient $\alpha_t \in [0, 1]$ mixes the two sources,

$$R_i^\bullet(\alpha_t) = \alpha_t R_{g,i}^\bullet + (1-\alpha_t) R_{d,i}^\bullet, \quad \bullet \in \{\text{resp},\ \text{token}\},$$

which is the blend carried out in Algorithm 1. The sequence-level advantage is the group normalization of the blended response reward across the $G$ sampled responses,

$$\hat{A}_i(\alpha_t) = \frac{R_i^{\text{resp}}(\alpha_t) - \text{mean}\big(\{R_j^{\text{resp}}(\alpha_t)\}_{j=1}^G\big)}{\text{std}\big(\{R_j^{\text{resp}}(\alpha_t)\}_{j=1}^G\big)},$$

the same group-relative estimator GRPO and GSPO use (Appendix J). Every run in this paper sets the default strategy to $S_{\text{def}} = \texttt{"zero"}$, so $R_d^{\text{token}} = \mathbf{0}$ and the token term collapses to $R_i^{\text{token}}(\alpha_t) = \alpha_t\,\mathbf{b}_i \cdot \mathbf{r}$, the correction written above. The $\alpha_t$ multiplying $\mathbf{b}_i \cdot \mathbf{r}$ and the $\alpha_t$ weighting the guided reward are therefore the same coefficient,

and the three-phase schedule of Figure 1 moves both at once.

The division by $|y_i|$ puts the two terms on one scale. $\hat{A}_i$ scores a response as a whole, so adding it directly to a per-token correction would let the sequence-level contribution grow with response length while the rubric contribution stayed fixed. Spreading it uniformly gives each of the $|y_i|$ tokens the share $\hat{A}_i/|y_i|$, and the rubric term then raises or lowers individual spans around that per-token baseline. The sequence-level importance ratio $s_i(\theta)$ carries the matching $1/|y_i|$ exponent, so ratio and advantage are normalized at the same granularity.

Before the rubric term is applied, each sampled response is routed by a verifiable correctness check: the choice implied by $y_i$ is compared against the ground-truth option, and the resulting correctness flag — together with the current $\alpha_t$ — decides whether the response is sent to $J_\phi$ for token-level rubric evaluation or falls through to the default reward. This is a routing operation over the sampled group, not a learned module; it is the ✓ correct / × incorrect branch drawn in Figure 1 and the corresponding branch of Algorithm 1. Its sign convention is what allows the same criterion score to add credit on a correct response and subtract it on an incorrect one.

**Dynamic Evolutionary Strategy.** Specifically, we implement $\alpha$ as a time-dependent dynamic coefficient $\alpha_t$ to balance rapid rule acquisition and general reasoning. The training process consists of three phases: (1) a **fully-guided phase** where $\alpha_t = 1$, utilizing dense token-level feedback to accelerate cold-start convergence; (2) a **linear decay phase** where $\alpha_t$ linearly decreases to 0, gradually shifting the model from strict constraint adherence to autonomous optimization; and (3) a **free exploration phase** where $\alpha_t = 0$, allowing the policy to refine general reasoning paths without scaffolded constraints.

**Criteria design and LLM judgments.** To address the complex, multi-dimensional evaluation demands of personality alignment tasks, we establish a structured checklist-style rubric evaluation system. Specifically, we formalize our rubric $\mathcal{C}$ as a set of n multi-dimension verifiable criteria: $\mathcal{C} = \{c_1, ..., c_n\}$. Each criterion is specified by a clear description to the overall evaluation. Our criteria design and judge prompt template for the grader is provided in Appendix G.

For a single criterion evaluation, the grader function $J_\phi$ implemented by a LLM takes the judge prompt as input and outputs a score $b_k = J_\phi(y_i, x, c_k) \in \{-1, 0, 1\}$. When the response is incorrect, the tokens of the sentence cited for $c_k$ have $b_k \in \{-1, 0\}$. Otherwise, tokens have $b_k \in \{0, 1\}$. Extending this to the full rubric, the grader $J_\phi$ evaluates all criteria and produces a vector $\mathbf{b}_i = J_\phi(y_i, x, \mathcal{C}) = [b_1^i, b_2^i, ..., b_n^i]$ where each $b_k^i$ represents the satisfaction score of criterion $c_k$. Each criterion carries a *criterion weight* $\mathbf{r} = [r_1, ..., r_n]$: a fixed, hand-set vector declared once at initialization and never updated during training. The score itself comes from the external grader $J_\phi$; $\mathbf{r}$ only sets how much each criterion contributes. In the released configuration the criteria are weighted equally, $r_k = 1/n$: the five personality-alignment criteria give $\mathbf{r} = [0.2, 0.2, 0.2, 0.2, 0.2]$, and the four-criterion Medical and Instruction-Following rubrics of Appendix G give $r_k = 0.25$.

The correctness-gated routing of the previous paragraph leverages the stability benefits of sequence-level updates for high-quality responses while retaining localized token-level feedback for responses requiring detailed correction. In our experiments, this improves training stability and alignment accuracy.

**Token Localization Strategy.** To translate sentence-level judgments into fine-grained token-level signals, we employ a hierarchical localization mechanism to map the evaluated sentence $s$ back to its corresponding token indices in the generated response $y_i$. First, we prioritize **Exact Tokenizer Matching**, attempting to locate the specific subsequence of token IDs in $y_i$ that matches the encoded ids of $s$. If this fails due to tokenization artifacts, we fall back to **Fuzzy Text Matching**, where we locate the character span of $s$ within the decoded text of $y_i$ and project it back to the corresponding token indices. Finally, where localization is infeasible (e.g., structural hallucination), we apply a **Uniform Fallback** strategy, distributing the criterion score across the entire sequence to preserve a training signal. This tier accounts for 4.3–15.5% of criterion evaluations depending on the domain (Appendix F). This multi-stage approach makes the reward vector $\mathbf{b}_i$ more spatially localized, applying credit or penalty primarily to the spans associated with the specific criterion. Detailed pseudocode of GSSPO is shown in Appendix K; Appendix F measures how often each tier fires and shows that localization uncertainty does not destabilize train-

ing.

## 4 Experiments

To rigorously evaluate the effectiveness and underlying mechanisms of GSSPO, we structure our experiments to answer four core research questions:

- **RQ1:** How does GSSPO compare with established personality-alignment baselines across varying model scales, particularly in challenging *SFT-Free* settings?
- **RQ2:** Is fine-grained token-level supervision superior to sequence-level signals, and how does the evolutionary curriculum impact learning efficiency?
- **RQ3:** How do different reward shaping strategies influence optimization stability? Does GSSPO reduce the gradient-conflict behavior observed under multi-criterion reward shaping?
- **RQ4:** Does GSSPO's granularity-aware optimization generalize beyond personality alignment to other verifiable domains such as medical reasoning and instruction following?

### 4.1 Experimental Setups

**Models and Training Settings.** To evaluate the scalability of our approach, we employ the Qwen3 model family (Yang et al., 2025) across varying parameter scales, specifically Qwen3-1.7B and Qwen3-8B. All training implementations are conducted using the ROLL framework (Wang et al., 2025). GSSPO and Rubric-based RL are all conducted in sequence-level importance sampling method. Comprehensive hyperparameters and experimental configurations are detailed in Appendix A.

**Evaluation Benchmarks.** We evaluate our method using three custom Choice Alignment (CA) benchmarks: **PRISM-CA**, **Big5Chat-CA**, and **Twin-2K-CA**. PRISM-CA and Big5Chat-CA are constructed by applying our PACF methodology to the PRISM (Kirk et al., 2024) and Big5Chat (Li et al., 2025) corpora. Twin-2K-CA is instead built directly from Twin-2K-500 (Toubia et al., 2025): rather than synthesizing distractors, we keep the original multiple-choice survey items—using their real answer options and each participant's recorded response as ground truth—retaining only blocks with sufficient label diversity and balancing labels within each block. PRISM-CA and Big5Chat-CA target multi-turn dialogue with explicit persona profiles; Twin-2K-CA stresses profile-to-decision inference, where each instance pairs a rich participant profile (demographics, Big 5, cognitive abilities, economic preferences) with a behavioral decision scenario drawn from the Mega-Study. Each instance is formatted as a multiple-choice problem with exactly one correct option: PRISM-CA and Big5Chat-CA use 4–7 options (the human-preferred response plus calibrated distractors), while Twin-2K-CA inherits each survey item's native option count (2–7). Table 1 summarizes the splits; comprehensive details regarding dataset statistics and construction are provided in Appendix H.

| Dataset | Training | Validation | Test |
|---|---|---|---|
| **PRISM-CA** | 5704 | 545 | 5798 |
| **Big5Chat-CA** | 2397 | 100 | 540 |
| **Twin-2K-CA** | 3302 | 825 | 1658 |

Table 1: Statistics of our three Choice-Alignment benchmarks. Twin-2K-CA test set uses the held-out-persona split for true profile-level generalization.

**Baselines.** We compare GSSPO against three representative categories of baseline methods: **(1) Standard Baselines:** This group includes the pre-trained base model (evaluated in a zero-shot setting), standard Supervised Fine-Tuning (**SFT**), and the classical **PPO** algorithm (Schulman et al., 2017). **(2) SFT-Free RL:** This category comprises reinforcement learning algorithms applied directly to the base model without a preceding SFT warm-up phase. We include **GRPO** (Shao et al., 2024) and **GSPO** (Zheng et al., 2025), which utilize fixed rule-based reward functions, as well as **Rubric-Based RL**, a baseline we construct by turning rubric grading into a training reward: an LLM-as-a-judge scores criteria compliance in the style of the HealthBench rubric protocol (Arora et al., 2025), and the resulting score is applied as a uniform reward across the sequence. **(3) SFT+RL:** This group represents two-stage alignment strategies where the model is first fine-tuned via SFT and subsequently optimized using reinforcement learning.

| Method | Qwen3-1.7B | | | Qwen3-8B | | |
|---|---|---|---|---|---|---|
| | PRISM-CA | Big5Chat-CA | Twin-2K-CA | PRISM-CA | Big5Chat-CA | Twin-2K-CA |
| Base | 18.26 | 40.68 | 39.76 | 24.16 | 35.42 | 51.18 |
| SFT | 42.84 | 62.03 | 61.61 | 58.45 | 66.48 | 63.39 |
| PPO (Schulman et al., 2017) | 48.43 | 37.22 | 62.50 | 68.50 | 57.59 | 64.00 |
| DPO (Rafailov et al., 2023) | 51.95 | 58.33 | 40.34 | 69.16 | 73.89 | 50.02 |
| SimPO (Meng et al., 2024) | 51.43 | 53.70 | 36.36 | 71.05 | 70.93 | 49.33 |
| **SFT-Free RL** | | | | | | |
| GRPO (Shao et al., 2024) | 68.54 | 61.29 | 54.40 | 78.10 | 75.00 | 62.90 |
| GSPO (Zheng et al., 2025) | 65.03$_{-3.5}$ | 63.33$_{+2.0}$ | 60.30$_{+5.9}$ | 83.42$_{+5.3}$ | 77.41$_{+2.4}$ | 61.20$_{-1.7}$ |
| Rubric-based RL | 63.45$_{-5.1}$ | **69.07**$_{+7.8}$ | 48.84$_{-5.6}$ | 75.25$_{-2.8}$ | 79.07$_{+4.1}$ | 57.72$_{-5.2}$ |
| **GSSPO (ours)** | **76.52**$_{+8.0}$ | 67.78$_{+6.5}$ | **62.10**$_{+7.7}$ | **84.86**$_{+6.8}$ | **81.30**$_{+6.3}$ | **64.70**$_{+1.8}$ |
| **SFT + RL** | | | | | | |
| GRPO | 78.35 | 60.19 | 61.60 | 77.98 | 83.70 | 64.40 |
| GSPO | 79.68$_{+1.3}$ | 64.26$_{+4.1}$ | 60.10$_{-1.5}$ | 86.10$_{+8.1}$ | 83.52$_{-0.2}$ | 64.20$_{-0.2}$ |
| Rubric-based RL | 73.30$_{-5.0}$ | 68.70$_{+8.5}$ | 55.26$_{-6.3}$ | 78.99$_{+1.0}$ | 81.67$_{-2.0}$ | 62.42$_{-2.0}$ |
| **GSSPO (ours)** | **81.74**$_{+3.4}$ | **69.81**$_{+9.6}$ | **65.90**$_{+4.3}$ | **87.94**$_{+10.0}$ | **84.63**$_{+0.9}$ | **66.60**$_{+2.2}$ |

Table 2: **Main personality-alignment evaluation on Qwen3-1.7B and Qwen3-8B.** Acc@1 (%) is reported for PRISM-CA and Big5Chat-CA; distance accuracy (%) is reported for Twin-2K-CA, which scores ordinal proximity between the predicted and gold option. Base and SFT are non-RL references; GRPO, GSPO, Rubric-based RL, and GSSPO are evaluated in both SFT-Free RL and SFT+RL regimes. Within each RL regime block, the best result in each column is marked in bold. Subscripts on non-GRPO RL rows report percentage-point (pp) changes relative to **GRPO** within the same column and regime (green: gain, red: drop).

### 4.2 Performance Analysis (Answering RQ1)

We first address **RQ1** by analyzing the comparative results on all three CA benchmarks, as presented in Table 2.

**Performance in SFT-Free Settings.** GSSPO performs strongly in "cold-start" scenarios. Unlike traditional RLHF pipelines that often rely on an SFT warm-up, GSSPO can be applied directly to base models. On Qwen3-1.7B (PRISM-CA), GSSPO achieves **76.52%**, outperforming the strongest rule-based baseline in our comparison (GRPO, 68.54%) by **7.98** percentage points. This suggests that dense, fine-grained supervision provides a useful optimization signal even when the initial policy is weak.

**Integration with SFT+RL.** When integrated into the standard two-stage pipeline, GSSPO further improves over the SFT warm-start. On Qwen3-8B, GSSPO reaches **87.94%**, the best result among the evaluated methods. This provides evidence that GSSPO is complementary to supervised fine-tuning and can refine persona-relevant behavior beyond SFT alone.

**Twin-2K-CA: Stress Test on Profile-to-Decision Inference.** Twin-2K-CA differs structurally from PRISM-CA and Big5Chat-CA: each instance pairs a long participant profile with a behavioral decision scenario, making it a harder profile-to-decision inference setting. As shown in the right block of Table 2, GSSPO remains the strongest method on Twin-2K-CA in the SFT+RL setting. This suggests that the trends observed on dialogue-based CA benchmarks also extend to longer-context behavioral decision prediction, while the narrower margins at 8B indicate that the benchmark may be closer to saturation at larger scale.

### 4.3 Mechanism Dissection (Answering RQ2 & RQ3)

To answer **RQ2** and **RQ3**, we conduct extensive ablation studies on the Qwen3-1.7B model. Figure 2 consolidates these analyses.

**RQ2: Granularity and Curriculum Dynamics.** Figure 2(a) evaluates the effect of token-level granularity. The **Token-level Scaling** variant (red line) consistently outperforms sparse sequence-level signals across varying $\alpha$, suggesting that dense rewards provide more useful gradient information for alignment. Furthermore, Figure 2(b) supports the effectiveness of our evolutionary strategy. A balanced "Scaffold-then-Wean" curriculum achieves the highest accuracy in this ablation, avoiding both underfitting from overly short guidance and overfitting from prolonged guidance.

**RQ3: Reward Shaping and Stability.** We investigate how to assign the token-level sentence reward magnitude $V_{sr}$ based on global response quality. We compare four strategies: (1) **Symmetric** ap-

| Method | Medical | | | Instruction Following | | STEM | | |
|---|---|---|---|---|---|---|---|---|
| | MedQA | MedMCQA | PubMedQA | $\text{IFEval}_{\text{loose}}$ | $\text{IFBench}_{\text{strict}}$ | MATH-500 | MMLU-Pro | GPQA |
| Base | 65.59 | 61.51 | 39.60 | 55.85 | 16.00 | 62.00 | 38.20 | 35.40 |
| SFT | 68.11 | 59.02 | 64.20 | 72.80 | 18.00 | 72.00 | 45.30 | **49.00** |
| **SFT + RL** | | | | | | | | |
| GRPO | 73.29 | 61.30 | 66.80 | 71.70 | 35.00 | 69.00 | 49.90 | 45.50 |
| GSPO | 69.36$_{-3.9}$ | 61.80$_{+0.5}$ | 72.60$_{+5.8}$ | 77.00$_{+5.3}$ | 32.30$_{-2.7}$ | 77.00$_{+8.0}$ | 57.70$_{+7.8}$ | 38.90$_{-6.6}$ |
| Rubric-based RL | 69.36$_{-3.9}$ | 59.96$_{-1.3}$ | 72.70$_{+5.9}$ | 80.00$_{+8.3}$ | 40.70$_{+5.7}$ | 76.40$_{+7.4}$ | 59.80$_{+9.9}$ | 40.40$_{-5.1}$ |
| **GSSPO (ours)** | **73.68**$_{+0.4}$ | **62.35**$_{+1.1}$ | **73.30**$_{+6.5}$ | **83.50**$_{+11.8}$ | **44.30**$_{+9.3}$ | **78.60**$_{+9.6}$ | **60.90**$_{+11.0}$ | 42.40$_{-3.1}$ |

Table 3: **Cross-domain evaluation on Qwen3-8B-Base.** Acc@1 (%) is reported for Medical, Instruction Following, and STEM benchmarks. Base and *SFT* are non-RL references; GRPO, GSPO, Rubric-based RL, and GSSPO are SFT+RL variants initialized from *SFT*. The best result in each column is marked in bold. Subscripts on non-GRPO RL rows report percentage-point (pp) changes relative to **GRPO** within the same column (green: gain, red: drop).

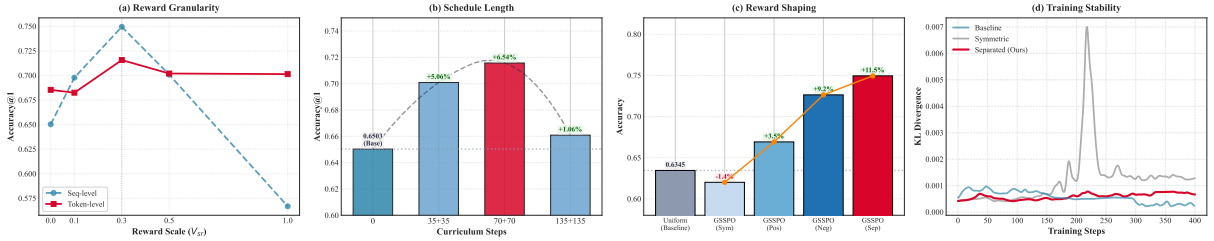



Figure 2: **Ablation Studies addressing RQ2 and RQ3. (a) & (b) address RQ2:** Token-level supervision ($\alpha = 0.3$) and a moderate scaffold-then-wean curriculum yield the best performance in our ablation. **(c) & (d) address RQ3:** The *Strictly Separated* reward shaping strategy ($+11.50$ pp over the Uniform baseline) reduces gradient conflict and maintains lower KL divergence (red line) compared to the *Symmetric* approach (gray line).

plies the full range $\{+V_{sr}, 0, -V_{sr}\}$ regardless of correctness; (2) **Positive-Biased** restricts rewards to $\{+V_{sr}, 0\}$ for correct responses; (3) **Negative-Biased** restricts penalties to $\{0, -V_{sr}\}$ for incorrect responses; and (4) **Strictly Separated (Ours)** completely disentangles feedback, assigning $\{+V_{sr}, 0\}$ exclusively to correct responses and $\{0, -V_{sr}\}$ to incorrect ones.

**Analysis.** As shown in Figure 2(c), the *Symmetric* strategy underperforms the Uniform baseline. The KL divergence spikes in Figure 2(d) suggest a gradient-conflict explanation. When a globally correct response contains specific errors, the symmetric approach applies conflicting signals (positive sequence reward vs. negative token penalty $-V_{sr}$), which can destabilize optimization. In contrast, our *Strictly Separated* strategy encourages **Sign Alignment** by aligning token-level feedback with the global update direction, achieving a **+11.50** pp gain over the Uniform baseline with more stable KL behavior. Detailed case study is shown in Appendix I.

### 4.4 Cross-Domain Generalization (Answering RQ4)

To evaluate GSSPO beyond personality alignment, we extend our framework to three out-of-domain settings on Qwen3-8B-Base: **Medical reasoning**, **Instruction Following (IF)**, and **STEM reasoning**. Each setting uses a domain-adapted SFT initializer followed by RL with the same family of objectives (GRPO/GSPO/Rubric-based RL/GSSPO); Appendix A.3 describes the corresponding verifiable RL corpora.

**GSSPO generalizes across domains.** Table 3 shows that GSSPO's gains are not limited to personality alignment in our experiments. On **Medical reasoning**, GSSPO attains the best accuracy on all three QA benchmarks (MedQA **73.68**, MedMCQA **62.35**, PubMedQA **73.30**), narrowly ahead of GRPO, GSPO, and Rubric-based RL. On **Instruction Following**, GSSPO reaches **83.50%** IFEval-loose and **44.30%** IFBench-strict, leading the strongest baseline in our comparison (Rubric-based RL) by $+3.5$ and $+3.6$pp and improving over the SFT cold-start by $+10.7$ and $+26.3$pp respectively. On **STEM reasoning**, GSSPO obtains the best MATH-500 and MMLU-Pro results (**78.60%**

and **60.90%**), while SFT remains strongest on GPQA-Diamond. Together these answer **RQ4**: the granularity-aware credit assignment behind GSSPO's personality-alignment gains can also improve performance on the evaluated verifiable out-of-domain tasks. These three domains test the generality of the credit-assignment mechanism, where correctness is externally verifiable. Construct validity for personality alignment itself rests on Twin-2K-CA, whose held-out-persona split requires profile-level generalization, and on the open-ended generation study of Appendix B.

### 4.5 Preference Baselines and Alignment Tax

**Preference-optimization baselines.** As reported in Table 2, on PRISM-CA with Qwen3-8B, sequence-level preference methods — **DPO** (Rafailov et al., 2023) (69.16%) and **SimPO** (Meng et al., 2024) (71.05%) — underperform GSSPO's 84.86%, with gaps of +15.70pp and +13.81pp respectively. This suggests that token-level credit assignment captures stylistic and persona-level cues that are less directly represented by pairwise preference labels.

**Alignment Tax on General Capabilities.** A common worry with rubric-based RL is that aggressive persona-targeted optimization may degrade general capability. We evaluate Qwen3-8B SFT+{GRPO, GSPO, GSSPO} on **Arena-Hard-Auto** (Li et al., 2024) and report retention relative to the SFT base. SFT+GSSPO retains **80.2%** of the SFT base's win-rate against the reference model, outperforming SFT+GRPO (77.4%) and SFT+GSPO (78.8%). In this evaluation, the dense, sentence-localized credit assignment in GSSPO improves alignment quality while retaining slightly more broad-coverage capability than the alternative sequence-level RL objectives.

## 5 Conclusion

In this paper, we introduced a two-stage framework for personality alignment in Large Language Models. We first presented the **Personality Alignment Choice Forge (PACF)**, a scalable pipeline that converts conversational data into choice-based benchmarks, simplifying the alignment task while enabling automatic evaluation grounded in observed human choices.

Building on this, we introduced **Group Sentence-Specified Sequence Policy Optimization (GSSPO)**, a reinforcement learning algorithm that leads every benchmark and scale in the SFT+RL regime and, among the RL methods compared, five of six in the SFT-Free regime. GSSPO uses a hybrid objective that combines sequence-level updates with fine-grained, token-level feedback. Our ablation studies support its core design principles, showing consistent benefits from token-level supervision, an evolutionary curriculum, and our reward shaping strategy.

Together, PACF and GSSPO offer a practical pathway toward developing more personally aligned language models under measurable preference-alignment objectives. Our findings show that localized reward assignment can improve personality-alignment performance over the evaluated baselines and provide evidence for its usefulness in reward engineering for large-scale generative models. Moreover, the framework may extend from simulating a single individual to studying larger-scale simulated populations, subject to careful validation.

## 6 Limitations

While our proposed framework improves personality alignment in our experiments, several limitations warrant consideration. First, the GSSPO algorithm's efficacy is fundamentally tethered to the quality and consistency of the "judger" LLM ($J_\phi$) responsible for providing rubric-based rewards. This dependency introduces a potential ceiling effect, as the trained model's alignment capabilities are ultimately bounded by the judger's evaluative prowess. Furthermore, deploying a powerful LLM as a judger incurs substantial computational overhead during training, as each generated sample requires an additional inference pass to compute the fine-grained reward signals. This could pose practical challenges for scalability and resource-constrained environments.

Second, the PACF pipeline, despite its automation, relies on a choice-based abstraction of personality. By reformulating open-ended dialogue into a multiple-choice task, we inherently simplify the complex, continuous nature of human expression. The quality of the resulting benchmark is contingent on the generated distractors, which may not always capture the full spectrum of plausible yet misaligned responses. This discrete formulation, while enabling objective evaluation and constraining the action space, may not fully address the nuances of generating aligned, free-form text

in real-world interactive settings. Future research could explore more cost-efficient reward modeling techniques and investigate hybrid methods that bridge the gap between discrete choice-based training and open-ended generation.

Third, GSSPO requires a profile at inference time. A user with no interaction history and no profile falls outside what the method conditions on, and the rubric criteria, each written against a profile, have nothing to refer to. True cross-population cold start is left to future work, through active preference elicitation or few-shot meta-personalization.

## 7 Ethical Considerations

While our work on **PACF** and **GSSPO** aims to enhance user experience in applications such as educational tutoring, role-play gaming, and compassionate healthcare support, we acknowledge the potential risks associated with high-fidelity personality simulation.

**Risk of Impersonation and Dual Use.** The capability to align LLMs closely with specific human profiles introduces the risk of non-consensual impersonation or the creation of deceptive "digital clones" for malicious social engineering. Deployments should be restricted to synthetic personas or to individuals who have consented to being simulated. Our own experiments use participants from datasets released for public research reuse; those releases cover reuse of the collected data rather than simulation of the participants, and we release no per-participant policy. Deployments of such systems should follow transparency standards so that users are aware they are interacting with a simulated agent.

**Bias Amplification.** Our method relies on both human datasets and LLM-based judges for reward assignment. Biases present in the source data (e.g., demographic skews in PRISM) or the latent biases of the "judge" models may be amplified during the reinforcement learning process. Consequently, the aligned models might exhibit unintentional stereotyping. We recommend that downstream applications incorporate safety guardrails and fairness evaluations before deployment.

**Privacy Preservation.** The PACF pipeline processes detailed user profiles. We add no personal data beyond what the source releases contain and we make no attempt at re-identification, which the PRISM terms of use prohibit. We did not run a dedicated PII audit: the profile sketch retains the demographic and trait attributes present in the source data. Two properties of the design limit exposure at deployment time. First, profile construction is a *one-time, offline* step run over existing historical data; it is not re-executed per deployment user, so no live user data enters the pipeline at serving time. Second, at inference the policy consumes only a profile and the current context, never the target label, so the observed human choice used to build a benchmark item is never required at deployment. Future work extending this pipeline to private data must ensure strict anonymization protocols are applied prior to the "profile sketch" stage, and should treat the stored profile itself as the sensitive artifact.

**Ethics Review.** The persona data used in this work comes from previously released public datasets, and we collected no new data from human participants. The human reference judgments of Appendix D were produced by paid annotators carrying out a bounded rating task against a published rubric: we recorded only their ratings and the evidence they cited, and we collected and analyzed no personal data about the annotators themselves.

**Use of AI Assistants.** We used AI assistants in this work. Coding assistants supported experiment design, scripting, and data processing, and a large language model assistant was used to polish the language of the paper. The research questions, the method, and all reported results are the authors' own, and the authors have verified and take responsibility for the content of this paper.

**Dataset Availability** Our benchmarks are derived artifacts, so we state the terms of the sources they are built from. PRISM (Kirk et al., 2024) is dual-licensed: human-written text under CC-BY-4.0 and model responses under CC-BY-NC-4.0, and its terms prohibit re-identification. PACF keeps the human-preferred response of the source dataset as the gold option, so the PRISM-CA gold options inherit the non-commercial term. Big5-Chat (Li et al., 2025) is released under the Apache-2.0 License; it is generated by instruction-tuned LLMs over a public dialogue corpus and is therefore also subject to the terms of the models that generated it. Twin-2K-CA is built from Twin-2K-500 (Toubia et al., 2025), released under the CC-BY-4.0 License. Distractors in PRISM-CA and Big5Chat-CA are model-generated and are therefore also sub-

ject to the generating providers' terms. We release all three benchmarks for non-commercial research use: this is required for PRISM-CA and applied uniformly by choice.

**Code Availability** All training runs use the open-source ROLL framework. Our contribution on top of it — the GSSPO reward worker implementing the localization and blending logic of Section 3 and Algorithm 1 — is released together with the PACF construction, training, and evaluation scripts and the configuration files for the runs we report, including the rubric prompts of Appendix G and the grader settings of Appendix A. Reproducing a training run requires API access to the rubric grader, and rollouts are sampled, so we expect reported numbers to be matched in trend rather than exactly. The code is available at `https://github.com/HKUST-FintechLab/GSSPO`.

## Acknowledgements

We thank the HKUST Fintech Lab and the Hong Kong University of Science and Technology for the computational resources that made this work possible. We also thank the anonymous reviewers for their constructive feedback, which substantially improved this paper.

## References

Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. 2025. Self-evolving multi-agent simulations for realistic clinical interactions. *Preprint*, arXiv:2503.22678.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. 2025. Healthbench: Evaluating large language models towards improved human health. *Preprint*, arXiv:2505.08775.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. *Preprint*, arXiv:2310.01377.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Xiachong Feng, Longxu Dou, and Lingpeng Kong. 2025. Reasoning does not necessarily improve role-playing ability. *Preprint*, arXiv:2502.16940.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Pengrui Han, Rafal Kocielnik, Peiyang Song, Ramit Debnath, Dean Mobbs, Anima Anandkumar, and R. Michael Alvarez. 2025. The personality illusion: Revealing dissociation between self-reports and behavior in LLMs. *arXiv preprint arXiv:2509.03730*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. *Preprint*, arXiv:2212.10403.

Jonas Hubotter, Frederike Lubeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. 2026. Reinforcement learning via self-distillation. *ArXiv*, abs/2601.20802.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Maik Larooij and Petter Törnberg. 2025. Do large language models solve the problems of agent-based modeling? a critical review of generative social simulations. *Preprint*, arXiv:2504.03274.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

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

Wenkai Li, Jiarui Liu, Andy Liu, Xuhui Zhou, Mona Diab, and Maarten Sap. 2025. Big5-chat: Shaping llm personalities through training on human-grounded data. *Preprint*, arXiv:2410.16491.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pages 124198–124235.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arXiv:2302.04761.

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

Ivan Sekulić, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. 2024. Reliable llm-based user simulator for task-oriented dialogue systems. *Preprint*, arXiv:2402.13374.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Olivier Toubia, George Z. Gui, Tianyi Peng, Daniel J. Merlau, Ang Li, and Haozhe Chen. 2025. Twin-2k-500: A dataset for building digital twins of over 2,000 people based on their answers to over 500 questions. *Preprint*, arXiv:2505.17479.

Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, Zichen Liu, Haizhou Zhao, Dakai An, Lunxi Cao, Qiyang Cao, Wanxi Deng, Feilei Du, Yiliang Gu, Jiahe Li, and 22 others. 2025. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *Preprint*, arXiv:2506.06122.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Eunseop Yoon, Hee Suk Yoon, SooHwan Eom, Gunsoo Han, Daniel Wontae Nam, Daejin Jo, Kyoung-Woon On, Mark A. Hasegawa-Johnson, Sungwoong Kim,

and Chang D. Yoo. 2024. Tlcr: Token-level continuous reward for fine-grained reinforcement learning from human feedback. *Preprint*, arXiv:2407.16574.

Chenguang Zhao, Meirewuti Habule, and Wei Zhang. 2025a. Large language models (llms) as research subjects: Status, opportunities and challenges. *New Ideas in Psychology*, 79:101167.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025b. A survey of large language models. *Preprint*, arXiv:2303.18223.

Weixiang Zhao, Xingyu Sui, Yulin Hu, Jiahe Guo, Haixiao Liu, Biye Li, Yanyan Zhao, Bing Qin, and Ting Liu. 2025c. Teaching language models to evolve with users: Dynamic profile modeling for personalized alignment. In *Advances in Neural Information Processing Systems*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

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.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. 2026. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning. *Preprint*, arXiv:2508.16949.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. Personality alignment of large language models. In *International Conference on Learning Representations*.

# A Detailed Experiment Settings

## A.1 Training Configurations.

This section provides detailed training configurations, as shown in Table 4. The training-time rubric grader $J_\phi$ is `gpt-5-mini-2025-08-07` for the three personality-alignment benchmarks (PRISM-CA, Big5Chat-CA, Twin-2K-CA) and `gpt-5.4` for the cross-domain settings (Medical, Instruction Following, STEM), queried at temperature 0.3 with a 2,048-token output budget and low reasoning effort. This is the grader that produces the rubric vector $\mathbf{b}_i$ during training; the GPT-5.2 judge of Appendix B is used only at evaluation time.

Table 4: GSSPO training configuration

| category | Configuration |
|---|---|
| **Backbone** | Model Precision: bfloat16 |
| | Attention Implementation: fa-2 |
| **Sampling** | Rollout Samples per Prompt: 8 |
| | Max New Tokens(Actor): 2048 |
| | Validation Sampling: Enabled |
| | Validation Temperature: 0.3 |
| | Top-p: 0.7, Top-k: 20 |
| **Training** | Learning Rate: $1 \times 10^{-6}$ |
| | Weight Decay: 0.0 |
| | Full Guidance: 1 epoch |
| | Guidance Decay: 2 epochs |
| | Free Exploration: 2 epochs |
| | Effective Batch Size: 64 |
| | Max Gradient Norm: 1 |
| | Training Epochs: 5 |
| **Hardware** | GPUs: 8 $\times$ H800 |

## A.2 Detailed Evaluation Settings

For the Qwen3 results of Table 2, the models under evaluation are Qwen3-1.7B and Qwen3-8B (Yang et al., 2025); Appendix E applies the same protocol to the Llama backbones. Our generation parameters are set to Temperature=0.01, Top-P=0.9 and Top-K=20 for every reported choice-accuracy evaluation, with the maximum output length configured as 2048 tokens; the open-ended study of Appendix B and the cross-domain evaluations use their own settings. The selected option is extracted from each model's output by deterministic string matching, without any LLM-as-judge scoring.

## A.3 Cross-domain RL Corpora

For the cross-domain experiments in Table 3, we construct all RL corpora from verifiable tasks. The Medical corpus contains 9.9K training examples and 100 validation examples sampled from MedQA-USMLE and MedMCQA, formatted as four-choice clinical questions with letter-valued ground truth. The IF corpus contains 9.9K training examples and 100 validation examples from filtered IFEval-style multi-constraint instruction-following prompts, where correctness is computed by rule-based constraint checkers. The STEM corpus contains 10K training examples and 200 validation examples, split evenly between MMLU-auxiliary multiple-choice questions and MegaScience textbook-reasoning problems.

The cross-domain training configuration matches Table 4 except for the guidance schedule and the generation length. Each cross-domain run trains for 250 steps rather than 5 epochs, allocated as Full Guidance 50 steps, Guidance Decay 50 steps, and the remaining 150 steps of Free Exploration, and raises the maximum number of new tokens from 2,048 to 3,072 to accommodate the longer clinical and multi-constraint responses. The rubric grader for these three domains is `gpt-5.4` (Appendix A). Rubric-judge agreement with human references in these domains is reported in Table 8 (seven-judge mean $\rho$: STEM 0.86, Medical 0.77, IFEval 0.74); that table does not include `gpt-5.4` itself, so it bounds the reliability of the rubric protocol in these domains rather than of this particular grader.

# B Open-Ended Generative Evaluation

To address the concern that multiple-choice evaluation (PACF) is insufficient – and partly explained by the dissociation between self-reported traits and behavior (Han et al., 2025) – we conduct pairwise open-ended evaluation on PRISM-CA, Big5Chat-CA, and Twin-2K-CA. We use **GPT-5.2** as the judge to evaluate the open-ended generations from Qwen3-8B GSSPO and seven baselines, applying position swapping to mitigate ordering bias. Each comparison is evaluated under two judge prompts: a domain-specific persona/reasoning-quality prompt and a generic Arena-Hard-style prompt. We sample 300 PRISM-CA, 540 Big5Chat-CA, and 300 Twin-2K-CA instances. For Twin-2K-CA, the judge evaluates reasoning quality rather than final multiple-choice correctness.

Table 5 and Figure 3 show that GSSPO transfers strongly from choice-based training to free-

| Domain | GSSPO vs. | Persona prompt | Generic prompt |
|---|---|---|---|
| PRISM-CA | GSPO | +31.5 | +44.2 |
| | GRPO | +28.7 | +39.8 |
| | PPO | +27.5 | +33.5 |
| | Rubric | +7.2 | +11.8 |
| | SFT | +3.8 | +1.7 |
| | DPO | −0.7 | −5.3 |
| | SimPO | −3.3 | −5.7 |
| Big5Chat-CA | PPO | +57.1 | +54.9 |
| | SFT | +9.9 | +7.0 |
| | GRPO | +0.4 | −0.4 |
| | DPO | −1.6 | −2.0 |
| | GSPO | −2.5 | −3.1 |
| | SimPO | −3.2 | −5.6 |
| | Rubric | −5.8 | −4.0 |
| Twin-2K-CA | GSPO | +79.0 | +53.2 |
| | GRPO | +52.7 | +28.0 |
| | PPO | +17.3 | +1.2 |
| | SFT | +8.5 | −3.5 |
| | Rubric | −1.8 | −14.2 |
| | DPO | −14.0 | −21.8 |
| | SimPO | −16.0 | −24.8 |

Table 5: **Pairwise open-ended evaluation on Qwen3-8B.** Values are Δpp = GSSPO win rate minus opponent win rate, so positive numbers favor GSSPO. Each comparison uses GPT-5.2 judge with position-swapped judgments.

form generation against on-policy RL baselines, especially on PRISM-CA and Twin-2K-CA. The per-domain Win/Tie/Loss panels make the same pattern easy to compare across domains and judge prompts: GSSPO's largest gains are against GRPO, GSPO, and PPO, while Big5Chat-CA is more saturated and exhibits smaller differences among the stronger RL and preference-optimization baselines. Against offline preference methods (DPO and SimPO), GSSPO is roughly tied on PRISM-CA and Big5Chat-CA and trails on Twin-2K-CA. To check for verbosity bias, we compute within-pair Pearson correlations between response-length difference and win/loss outcome; all per-pair correlations satisfy $|r| \leq 0.28$, suggesting these preferences are not explained by longer responses alone.

**Where Rubric-based RL leads.** GSSPO trails Rubric-based RL on open-ended quality in two domains, Big5Chat-CA ($-5.8$/$-4.0$ pp) and Twin-2K-CA ($-1.8$/$-14.2$ pp). Big5Chat-CA is the one where Rubric-based RL also surpasses GSSPO on choice accuracy, in the SFT-Free regime (Qwen3-1.7B: 69.07 vs. 67.78, Table 2). On Big5Chat-CA we attribute this to a small, saturated corpus (2,397 training items; median dialogue 213 tokens vs. 868 for PRISM-CA). Under full guidance ($\alpha = 1$), the dense sentence-level reward supplies a strong, low-variance gradient that fits the rubric-conforming reasoning template early in training: choice accuracy rises sharply and policy entropy drops quickly, indicating reduced exploration over reasoning trajectories. Rubric-based RL's uniform sequence-level reward leaves the reasoning process less constrained, so it explores a wider range of justifications and scores higher on open-ended quality. Twin-2K-CA separates differently: there GSSPO leads Rubric-based RL on choice accuracy (66.60 vs. 62.42, Table 2) while trailing it on open-ended quality under both judge prompts, by a far wider margin under the generic one ($-14.2$ vs. $-1.8$ pp).

## C SDPO Comparison

We further compare GSSPO with SDPO (Hubotter et al., 2026) under different effective signal-density settings. Here, $s$ denotes the effective signal density relative to GSSPO. GSSPO uses sparse token-localized feedback, while SDPO applies dense feedback whose per-update reward mass is scaled to three effective signal densities relative to GSSPO ($s\approx1$, $s\approx2$, and $s\approx10$); the corresponding optimizer settings are given in the released configuration files. Tables 6 and 7 report choice-evaluation accuracy.

Overall, GSSPO remains strongest on PRISM-CA and Twin-2K-CA at both model scales, while dense SDPO can be competitive on Big5Chat-CA. The SDPO trend is most monotonic on Qwen3-1.7B, where higher signal density improves PRISM-

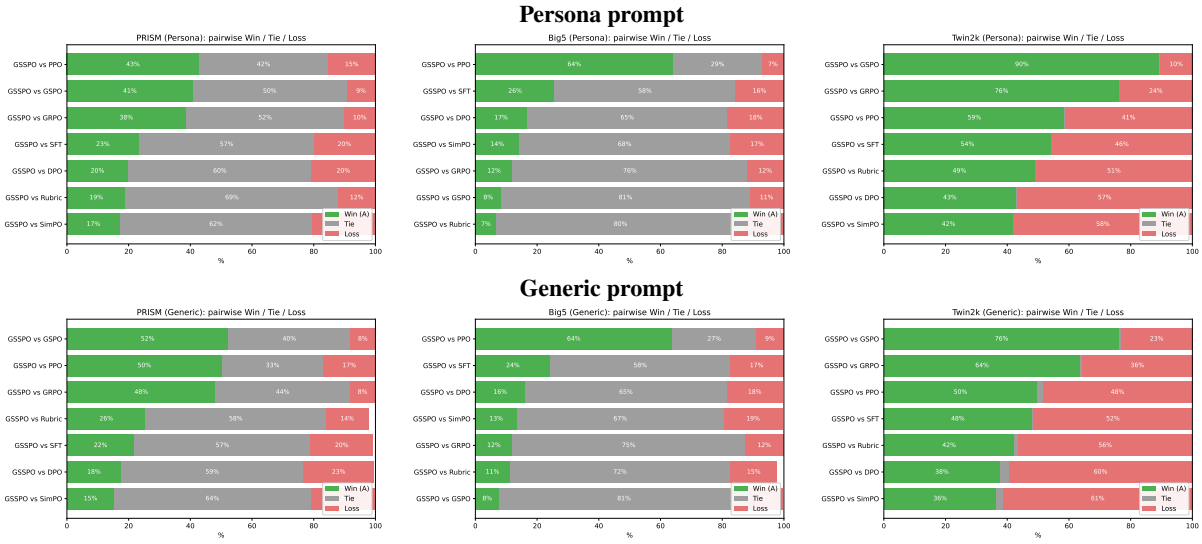



Figure 3: **Win/Tie/Loss for pairwise open-ended evaluation.** Each panel shows GSSPO's Win/Tie/Loss rates against each baseline for one domain (PRISM-CA, Big5Chat-CA, Twin-2K-CA). Top row: domain-specific persona/reasoning-quality prompt; bottom row: generic Arena-Hard-style prompt.

| Method | PRISM-CA | Big5Chat-CA | Twin-2K-CA |
|---|---|---|---|
| **GSSPO (best)** | **81.74** | 69.81 | **65.90** |
| SDPO ($s\approx1$) | 63.85 | 63.15 | 28.99 |
| SDPO ($s\approx2$) | 68.81 | 69.26 | 30.30 |
| SDPO ($s\approx10$) | 76.51 | **76.67** | 31.61 |

Table 6: **SDPO comparison on Qwen3-1.7B.** Accuracy (%) is reported on PRISM-CA, Big5Chat-CA, and Twin-2K-CA.

| Method | PRISM-CA | Big5Chat-CA | Twin-2K-CA |
|---|---|---|---|
| **GSSPO (best)** | **87.94** | **84.63** | **66.60** |
| SDPO ($s\approx1$) | 71.38 | 81.48 | 35.64 |
| SDPO ($s\approx2$) | 82.57 | 83.52 | 36.00 |
| SDPO ($s\approx10$) | 72.84 | 82.96 | 37.31 |

Table 7: **SDPO comparison on Qwen3-8B.** Accuracy (%) is reported on PRISM-CA, Big5Chat-CA, and Twin-2K-CA.

CA and Big5Chat-CA, but the 8B results show that density alone is not sufficient.

## D Cross-Domain LLM-Judge Agreement

Because GSSPO relies on rubric-based LLM judgments for reward assignment, we evaluate whether these judge signals agree with human reference annotations across domains. We sample 50 prompt-response pairs for each of four training methods in each domain, yielding 200 examples per domain and 1,200 examples in total. Each example is scored by seven LLM judges using the same ROLL/GSSPO judge prompts used during training, and we compare their scores and cited evidence against human reference judgments under the same rubric. The human reference judgments were produced by five students recruited at our institution, working from the same rubric and instructions given to the LLM judges, and paid HK$60 per hour. They were told that their judgments would be used to validate the automatic judges reported in this paper.

The seven judges are `gpt-5-mini-2025-08-07`, `gpt-5.1`, `gpt-5.2`, `claude-sonnet-4.6`, `claude-haiku-4.5`, `gemini-3.1-pro`, and `gemini-3-flash`. Among these, `gpt-5-mini-2025-08-07` is the training-time rubric grader $J_\phi$ for the three personality-alignment benchmarks (Appendix A); the cross-domain settings (Medical, Instruction Following, STEM) use `gpt-5.4` as $J_\phi$. This is distinct from the open-ended pairwise evaluation of Appendix B, which uses GPT-5.2 as an evaluation-time judge only.

The per-judge breakdown in Table 9 shows that the training grader is not a weak member of the panel: `gpt-5-mini-2025-08-07` attains the highest or joint-highest $\rho$ on PRISM (0.92), Big5 (0.99), Twin2K (0.74), and IFEval (0.83), and is mid-pack on STEM (0.89, third of seven) while falling slightly below the panel mean on Medical (0.74 vs. 0.77). Agreement broadly tracks model capability — `claude-haiku-4.5` is the weakest judge on five of the six domains (0.44–0.80; on IFEval `gpt-5.1` is lower at 0.58) — but capability is not the whole story: `claude-sonnet-4.6` is top-tier

| Domain | Mean Spearman $\rho$ | Exact agreement (%) | Evidence $F_1$ | Evidence cosine |
|---|---|---|---|---|
| PRISM | 0.84 | 90.3 | 0.79 | 0.972 |
| Big5 | 0.94 | 96.9 | 0.77 | 0.978 |
| Twin2K | 0.60 | 72.1 | 0.68 | 0.940 |
| STEM | 0.86 | 87.3 | 0.79 | 0.968 |
| IFEval | 0.74 | 81.4 | 0.61 | 0.912 |
| Medical | 0.77 | 82.9 | 0.83 | 0.984 |

Table 8: **Cross-domain LLM-judge agreement.** Seven LLM judges are compared against human reference judgments under the same domain-specific GSSPO rubrics. Mean Spearman $\rho$ measures per-criterion score rank correlation averaged over the seven judges; exact agreement compares discrete criterion scores; evidence $F_1$ and evidence cosine compare cited sentence evidence.

| Judge | PRISM | Big5 | Twin2K | STEM | IFEval | Medical |
|---|---|---|---|---|---|---|
| `gpt-5-mini-2025-08-07`$^\dagger$ | 0.92 | 0.99 | 0.74 | 0.89 | 0.83 | 0.74 |
| `gpt-5.1` | 0.83 | 0.93 | 0.59 | 0.83 | 0.58 | 0.71 |
| `gpt-5.2` | 0.80 | 0.90 | 0.56 | 0.87 | 0.72 | 0.78 |
| `claude-sonnet-4.6` | 0.80 | 0.99 | 0.48 | 0.87 | 0.71 | 0.77 |
| `claude-haiku-4.5` | 0.71 | 0.80 | 0.44 | 0.79 | 0.72 | 0.68 |
| `gemini-3.1-pro` | 0.88 | 0.95 | 0.71 | 0.91 | 0.82 | 0.82 |
| `gemini-3-flash` | 0.90 | 0.99 | 0.70 | 0.90 | 0.82 | 0.89 |
| **Seven-judge mean** | **0.84** | **0.94** | **0.60** | **0.86** | **0.74** | **0.77** |

Table 9: **Per-judge agreement with human references (Spearman $\rho$).** Row-wise breakdown of the mean reported in Table 8. $^\dagger$ `gpt-5-mini-2025-08-07` is the training-time rubric grader $J_\phi$ for the personality benchmarks; its row is therefore the agreement of the grader that actually shapes GSSPO's gradients, rather than a bystander judge.

on Big5 ($\rho = 0.99$) yet second-worst on Twin2K ($\rho = 0.48$, above only `claude-haiku-4.5`), so domain expertise matters independently of raw model strength. Twin2K is hard for every judge (best 0.74, worst 0.44), which is consistent with the difficulty of inferring individualized trait-to-decision behavior and bounds how much reward signal any grader can supply in that domain.

The agreement pattern is domain-dependent. Big5, STEM, and PRISM show relatively high agreement with the human references, with mean $\rho$ values of 0.94, 0.86, and 0.84, respectively. Twin2K is the hardest domain (mean $\rho = 0.60$), consistent with its richer profile-to-decision inference setting. These results suggest that rubric-based LLM judges largely agree with human references for dialogue-style personality alignment, while judge quality and domain expertise remain important limitations for harder behavioral-decision tasks.

## E Cross-Family Results on Llama

Table 10 repeats the SFT+RL comparison of Table 2 on Llama-3.2-3B and Llama-3.1-8B over the same three benchmarks. The Base and SFT rows give the starting point of each cell, so every RL result can be read against the model it was trained from.

GSSPO is the strongest method in every column. The result holds at both model sizes and on all three benchmarks, which locates the advantage in the credit-assignment objective rather than in a particular backbone or corpus.

Twin-2K-CA draws the sharpest distinction. It is the only benchmark here on which an RL objective can finish below the SFT model it started from, and at 8B all three baselines do, while GSSPO improves on its starting point. At 3B the spread narrows and only Rubric-based RL ends below its reference. Held-out-persona evaluation rewards localized credit and exposes uniform sequence-level credit; Appendix F reports how reliably the localization step that supplies that credit fires.

## F Token Localization Reliability

Section 3 maps each judged sentence back to token spans through a three-tier cascade — exact tokenizer matching, fuzzy text matching, and a uniform fallback. Because every token-level reward depends on that mapping, we measure how often each tier fires and whether localization uncertainty destabilizes training.

**Tier frequency.** We instrument the localization module to log the tier used for every criterion eval-

| Method | Llama-3.2-3B | | | Llama-3.1-8B | | |
|---|---|---|---|---|---|---|
| | PRISM-CA | Big5Chat-CA | Twin-2K-CA | PRISM-CA | Big5Chat-CA | Twin-2K-CA |
| Base | 25.35 | 34.26 | 56.99 | 25.16 | 34.44 | 57.72 |
| SFT | 59.95 | 62.78 | 60.78 | 69.06 | 72.22 | 63.75 |
| **SFT + RL** | | | | | | |
| GRPO | 79.01 | 79.81 | 61.11 | 86.18 | 86.85 | 59.42 |
| GSPO | 83.94$_{+4.9}$ | 76.67$_{-3.1}$ | 61.22$_{+0.1}$ | 85.24$_{-0.9}$ | 83.89$_{-3.0}$ | 54.95$_{-4.5}$ |
| Rubric-based RL | 81.42$_{+2.4}$ | 78.33$_{-1.5}$ | 59.42$_{-1.7}$ | 86.36$_{+0.2}$ | 87.78$_{+0.9}$ | 62.78$_{+3.4}$ |
| **GSSPO (ours)** | **87.55**$_{+8.5}$ | **81.85**$_{+2.0}$ | **63.34**$_{+2.2}$ | **89.96**$_{+3.8}$ | **88.89**$_{+2.0}$ | **65.78**$_{+6.4}$ |

Table 10: **Cross-family personality alignment on Llama-3.2-3B and Llama-3.1-8B (SFT+RL).** Acc@1 (%) is reported for PRISM-CA and Big5Chat-CA; distance accuracy (%) is reported for Twin-2K-CA. Base and SFT are non-RL references; GRPO, GSPO, Rubric-based RL, and GSSPO are evaluated in the SFT+RL regime. The best result in each column is marked in bold. Subscripts on non-GRPO RL rows report percentage-point (pp) changes relative to **GRPO** within the same column (green: gain, red: drop).

uation over the same sample as Appendix D (1,200 examples, six domains); Table 11 reports the resulting tier frequencies.

| Domain | Exact (%) | Fuzzy (%) | Fallback (%) |
|---|---|---|---|
| PRISM | 42.5 | 45.0 | 12.5 |
| Big5 | 32.0 | 60.6 | 7.4 |
| Twin2K | 32.2 | 52.4 | 15.5 |
| STEM | 26.2 | 67.5 | 6.3 |
| Medical | 26.2 | 59.5 | 14.3 |
| IFEval | 80.2 | 15.5 | 4.3 |

Table 11: **Token-localization tier frequency**, measured over the Appendix D sample.

Exact or fuzzy matching resolves 84.6%–95.7% of criterion evaluations across the six domains, and the uniform fallback is confined to 4.3%–15.5%, concentrated on the longer, paraphrase-heavy behavioral and medical domains. IFEval is the opposite extreme: its verbatim, constraint-structured responses push exact matching to 80.2%.

**Does localization uncertainty destabilize training?** On the two runs carrying per-step localization telemetry (Instruction Following and Medical) we correlate the per-step fuzzy-plus-fallback rate $f_t$ with the tracked stability diagnostics, over every step in which localization is active — the full-guidance and decay phases, i.e. the entire population of steps where a fallback can occur — pooled with per-run $z$-scoring.

As Table 12 shows, the divergence and variance diagnostics all fall as localization uncertainty rises, while policy entropy moves the other way ($+0.50$/$+0.49$). Token-reward variance here is the token-level reward std$^2$; the group-normalized, clipped advantage is range-bounded by construction and is not tracked separately. A first-difference

| Correlate of $f_t$ | Pearson $r$ | Spearman $\rho$ |
|---|---|---|
| KL to old policy (magnitude) | $-0.24$ | $-0.26$ |
| KL penalty to reference | $-0.58$ | $-0.59$ |
| Token-reward variance | $-0.37$ | $-0.15$ |
| Policy entropy | $+0.50$ | $+0.49$ |

Table 12: **Localization-uncertainty rate $f_t$ vs. training-stability diagnostics** (two instrumented GSSPO runs, run-wise $z$-scored). All four Pearson correlations are significant: $p < 0.01$ for KL to old policy and $p < 0.001$ for the other three.

test shows that fallback spikes do not co-move with KL or variance spikes (all $|\Delta r| < 0.17$, n.s.), ruling out spike-coupled instability beyond shared training trends.

The pattern matches the design rather than contradicting it. Precise localization concentrates credit on judge-cited spans, and the policy takes more decisive, lower-entropy steps; when localization degrades to the uniform fallback, the same correction spreads evenly and the update behaves like GSPO — smaller KL, lower reward variance, higher entropy. Localization uncertainty therefore moves GSSPO gracefully toward its GSPO limit instead of destabilizing training. For final performance, the localized-versus-uniform contrast is already visible in Table 2: Rubric-based RL applies a uniform sequence-level reward where GSSPO applies localized token-level corrections.

## G Criteria design and Prompt Template

Two grader prompts appear in this work and are used in different places. The **GSSPO prompt** returns a per-criterion score vector *and* the evidence sentence supporting each score, because the sentence is what the localization module maps back

to token spans; it is used whenever GSSPO is the policy being trained. The **Rubric-Based prompt** returns only a scalar quality score, with no evidence sentences, and is used for the Rubric-based RL baseline, whose reward is applied uniformly over the whole sequence. Both prompts are given the same rubric $\mathcal{C}$, so the two methods differ in reward granularity rather than in the criteria they are judged against. The open-ended pairwise judge of Appendix B is a third, evaluation-only prompt and is never used to produce training reward.

For our personality alignment task, we design rubric criteria to evaluate response quality:

- $c_1$: *Persona Consistency* - Does the response check if the choices maintain the character's personality, values, and behavioral patterns?
- $c_2$: *Conversational Coherence* - Does the response check if the choices are contextually appropriate and follow naturally from the dialogue?
- $c_3$: *Tone and Style Match* - Does the response check if the choices match the expected communication style and emotional tone?
- $c_4$: *Contextual Grounding* - Does the response show understanding of the specific situation and context?
- $c_5$: *Affective Alignment* - Does the response check if the choices appropriately reflect emotions and stance consistent with the persona?

Our Grader LLM prompt template is as Follows.

**Prompt for grader model (GSSPO)**

```
You are an evaluator for a next-turn
    prediction task.

GOAL
Given a PERSONA + multi-turn DIALOGUE (in
    PROMPT), a list of OPTIONS (for context)
    , the CORRECT_OPTION (for context), and
    a MODEL_OUTPUT (may include <THINK>...</
    THINK> plus the final choice), evaluate
    the MODEL_OUTPUT against specific
    CRITERIA.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
2) For each criterion with score!=0, provide
     exact sentence evidence from
    MODEL_OUTPUT ({neg_score_note})
   - Extract exact sentences that demonstrate
       the criterion
   - Extracted sentences should be some
       analysis or reasoning in the
       MODEL_OUTPUT that supports your score,
        don't extract sentences from the
       PROMPT or OPTIONS or some generic
       sentence that doesn't support your
       score
   - Use "ALL" if the entire response
       satisfies the criterion
   - Provide multiple sentences if applicable
3) Focus on quality assessment, NOT
    correctness of the final choice
4) Consider ALL text in MODEL_OUTPUT
    including reasoning in <THINK> tags
5) Output valid JSON only, no additional
    text

CRITERIA (score range: {scores_info}):
[1] Persona Consistency: Does the response
    check if the choices maintain the
    character's personality, values, and
    behavioral patterns?
[2] Conversational Coherence: Does the
    response check if the choices are
    contextually appropriate and follow
    naturally from the dialogue?
[3] Tone & Style Match: Does the response
    check the expected communication style
    and emotional tone?
[4] Contextual Grounding: Does the response
    show understanding of the specific
    situation and context?
[5] Affective Alignment: Does the response
    check if the choices appropriately
    reflect emotions and stance consistent
    with the persona?

OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
        score_info_identifier}, "sentences":
         [ "Exact sentence from MODEL_OUTPUT
        ", "Another sentence", ... ]}},
    {{"id": 2, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 3, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 4, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 5, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}}
  ],
  "explanation": "Optional brief explanation
      "
}}

CONTEXT
CORRECT_OPTION (for reference): {co_str}

PROMPT (PERSONA + DIALOGUE + OPTIONS):
{prompt}

MODEL_OUTPUT (to evaluate):
```

```
{response}
```

**Prompt for grader model(Rubric-Based)**

```
You are an evaluator for a next-turn
    prediction task.

GOAL
Given a PERSONA + multi-turn DIALOGUE (in
    PROMPT), a list of OPTIONS (for context)
    , the CORRECT_OPTION (for context), and
    a MODEL_OUTPUT (may include <THINK>...</
    THINK> plus the final choice), evaluate
    the MODEL_OUTPUT against specific
    CRITERIA.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
2) {neg_score_note}
3) Focus on quality assessment, NOT
    correctness of the final choice
4) Consider ALL text in MODEL_OUTPUT
    including reasoning in <THINK> tags
5) Output valid JSON only, no additional
    text

CRITERIA (score range: {scores_info}):
[1] Persona Consistency: Does the response
    check if the choices maintain the
    character's personality, values, and
    behavioral patterns?
[2] Conversational Coherence: Does the
    response check if the choices are
    contextually appropriate and follow
    naturally from the dialogue?
[3] Tone & Style Match: Does the response
    check the expected communication style
    and emotional tone?
[4] Contextual Grounding: Does the response
    show understanding of the specific
    situation and context?
[5] Affective Alignment: Does the response
    check if the choices appropriately
    reflect emotions and stance consistent
    with the persona?


OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
        score_info_identifier}}},
    {{"id": 2, "score": {
        score_info_identifier}}},
    {{"id": 3, "score": {
        score_info_identifier}}},
    {{"id": 4, "score": {
        score_info_identifier}}},
    {{"id": 5, "score": {
        score_info_identifier}}}
  ],
  "explanation": "Optional brief explanation
      "
}}
```

```
CONTEXT
CORRECT_OPTION (for reference): {co_str}

PROMPT (PERSONA + DIALOGUE + OPTIONS):
{prompt}

MODEL_OUTPUT (to evaluate):
{response}
```

### G.1 Criteria and Grader Prompt for Twin-2K-CA

For the Twin-2K-CA benchmark, the task is *behavioral decision prediction* given a rich person profile (demographics, Big 5, cognitive abilities, economic preferences, and qualitative self-description), rather than multi-turn dialogue. We therefore adapt the criteria to this decision-prediction setting, preserving the judger's freedom to extract evidence sentences while requiring that cited information be *correctly used*, not merely mentioned.

- $c_1$: *Demographic Grounding* — Does the reasoning reasonably reflect the person's demographic circumstances (income, age, education, political views, region, etc.)? Mere mention is insufficient; the direction of the cited attribute should be consistent with commonsense interpretation.
- $c_2$: *Trait-Based Reasoning* — Does the reasoning correctly employ the person's measured traits (Big 5 scores, cognitive abilities such as CRT, behavioral scales such as risk aversion, loss aversion, empathy)? The cited trait must be (a) relevant to the scenario, (b) interpreted in the correct direction, and (c) lead to a conclusion consistent with its semantic meaning. Irrelevant or inverted usage is treated as invalid evidence.
- $c_3$: *Scenario Comprehension* — Does the reasoning accurately grasp the decision scenario's structure (risk vs. reward, framing, anchoring, probabilistic trade-off, fairness stakes)? Not only recognizing the scenario type, but also correctly interpreting its numerical/logical relationships.
- $c_4$: *Profile-to-Decision Inference* — Is the chain from profile to the selected option ("because this person has X, they would choose Y") logically valid? If X actually supports the opposite option, or if there is an obvious leap between X and Y, the chain is marked invalid.

- $c_5$: *Individualized Reasoning* — Is the reasoning about *this specific individual* rather than the general population? Phrases such as "most people..." or "the average respondent..." are treated as generic reasoning and penalized even when the direction is correct.

### Prompt for grader model (GSSPO, Twin-2K-CA)

```
You are an evaluator for a behavioral
    decision-prediction task.

GOAL
Given a PERSON PROFILE and a DECISION
    SCENARIO (in PROMPT), a list of OPTIONS
    (for context), the CORRECT_OPTION (for
    context), and a MODEL_OUTPUT (may
    include <THINK>...</THINK> plus the
    final choice), evaluate the MODEL_OUTPUT
     against specific CRITERIA.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
2) For each criterion with score!=0, provide
     exact sentence evidence from
    MODEL_OUTPUT ({neg_score_note})
   - Extract exact sentences that demonstrate
       the criterion
   - A sentence counts as positive evidence
       only if the cited profile attribute
       or trait is used CORRECTLY: (a)
       relevant to the scenario, (b) in the
       correct direction, (c) consistent
       with the conclusion. Merely
       mentioning an attribute does not
       count.
   - Do not extract sentences from the PROMPT
       or OPTIONS, only from MODEL_OUTPUT
       reasoning
   - Use "ALL" if the entire response
       satisfies the criterion
   - Provide multiple sentences if applicable
3) Focus on quality of reasoning, NOT
    correctness of the final choice
4) Consider ALL text in MODEL_OUTPUT
    including reasoning in <THINK> tags
5) Output valid JSON only, no additional
    text

CRITERIA (score range: {scores_info}):
[1] Demographic Grounding: Does the
    reasoning reasonably and correctly
    reflect the person's demographic
    circumstances (income, age, education,
    political views, region, etc.)? Cited
    direction must match common sense.
[2] Trait-Based Reasoning: Does the
    reasoning correctly employ the person's
    measured traits (Big 5, cognitive
    abilities, economic/behavioral scales)?
    Require relevance to scenario, correct
    directional interpretation, and
    consistent conclusion. Irrelevant or
    inverted usage does not count.
[3] Scenario Comprehension: Does the
    reasoning accurately grasp the decision
    scenario's structure (risk/reward,
    framing, anchoring, probability trade-
    off, fairness stakes), including correct
     interpretation of numerical/logical
    relationships?
[4] Profile-to-Decision Inference: Is the
    reasoning chain from profile to selected
     option logically valid? ''Because X,
    therefore Y'' requires X to actually
    support Y; inverted support or obvious
    leaps are invalid.
[5] Individualized Reasoning: Is the
    reasoning about this specific individual
    , avoiding generic phrases like ''most
    people...'' or ''the average respondent
    ...''? Generic reasoning is penalized
    even when direction is correct.

OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
        score_info_identifier}, "sentences":
         [ "Exact sentence from MODEL_OUTPUT
        ", "Another sentence", ... ]}},
    {{"id": 2, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 3, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 4, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 5, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}}
  ],
  "explanation": "Optional brief explanation
      "
}}

CONTEXT
CORRECT_OPTION (for reference): {co_str}

PROMPT (PROFILE + SCENARIO + OPTIONS):
{prompt}

MODEL_OUTPUT (to evaluate):
{response}
```

### Prompt for grader model (Rubric-Based, Twin-2K-CA)

```
You are an evaluator for a behavioral
    decision-prediction task.

GOAL
Given a PERSON PROFILE and a DECISION
    SCENARIO (in PROMPT), a list of OPTIONS
    (for context), the CORRECT_OPTION (for
    context), and a MODEL_OUTPUT (may
    include <THINK>...</THINK> plus the
    final choice), evaluate the MODEL_OUTPUT
```

```
     against specific CRITERIA.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
2) {neg_score_note}
3) A criterion is satisfied only if the
    cited information is used CORRECTLY (
    relevant to scenario, in correct
    direction, consistent with the
    conclusion); merely mentioning an
    attribute does not count.
4) Focus on quality of reasoning, NOT
    correctness of the final choice
5) Consider ALL text in MODEL_OUTPUT
    including reasoning in <THINK> tags
6) Output valid JSON only, no additional
    text

CRITERIA (score range: {scores_info}):
[1] Demographic Grounding: Does the
    reasoning reasonably and correctly
    reflect the person's demographic
    circumstances?
[2] Trait-Based Reasoning: Does the
    reasoning correctly employ the person's
    measured traits with correct directional
     interpretation?
[3] Scenario Comprehension: Does the
    reasoning accurately grasp the decision
    scenario's structure and numerical/
    logical relationships?
[4] Profile-to-Decision Inference: Is the
    reasoning chain from profile to selected
     option logically valid?
[5] Individualized Reasoning: Is the
    reasoning about this specific individual
     rather than the general population?

OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
        score_info_identifier}}},
    {{"id": 2, "score": {
        score_info_identifier}}},
    {{"id": 3, "score": {
        score_info_identifier}}},
    {{"id": 4, "score": {
        score_info_identifier}}},
    {{"id": 5, "score": {
        score_info_identifier}}}
  ],
  "explanation": "Optional brief explanation
      "
}}

CONTEXT
CORRECT_OPTION (for reference): {co_str}

PROMPT (PROFILE + SCENARIO + OPTIONS):
{prompt}

MODEL_OUTPUT (to evaluate):
{response}
```

### G.2 Criteria and Grader Prompt for STEM

For the STEM corpus the task is *scientific problem solving* across mathematics, physics, chemistry, and related quantitative sciences. Outcome correctness is verified separately by matching the model's final or boxed answer against the ground-truth answer, allowing algebraically equivalent forms; that binary signal is what enters $R^{\text{resp}}$ through the correctness routing of Section 3. The five criteria below score the *reasoning process* and supply the token-level term $\alpha \mathbf{b}_i \cdot \mathbf{r}$, so a response can be penalized on a faulty intermediate step even when its final answer is right, and credited for a sound setup even when it is wrong.

- $c_1$: *Problem Setup & Variable Identification* — Does the response correctly identify variables, given conditions, constraints, and what is being asked?
- $c_2$: *Concept/Theorem Selection* — Does the response invoke the correct principle, theorem, formula, or domain knowledge for this problem?
- $c_3$: *Logical Reasoning* — Does each intermediate reasoning step logically follow from the prior steps and stated assumptions?
- $c_4$: *Symbolic/Numeric Computation* — Are the arithmetic, algebraic manipulation, equation solving, or numeric computations performed correctly?
- $c_5$: *Final Answer Correctness* — Does the final stated or boxed answer match the GROUND TRUTH ANSWER (allowing equivalent algebraic forms)?

**Prompt for grader model (GSSPO, STEM)**

```
You are an evaluator for a scientific
    reasoning task (mathematics, physics,
    chemistry, or related quantitative
    sciences).

GOAL
Given a PROBLEM, the GROUND TRUTH ANSWER,
    and a MODEL_OUTPUT (which may include <
    THINK>...</THINK> reasoning plus the
    final solution), evaluate the
    MODEL_OUTPUT against specific CRITERIA.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
```

```
2) For each criterion with score!=0, provide
     exact sentence evidence from
     MODEL_OUTPUT ({neg_score_note}):
   - Extract exact verbatim sentences from
       MODEL_OUTPUT that demonstrate the
       criterion. Do NOT paraphrase. Do NOT
       extract sentences from the PROBLEM or
        the GROUND TRUTH.
   - Use "ALL" if the entire response
       uniformly satisfies the criterion (
       rare).
   - Provide multiple sentences if applicable
       .
   - Skip filler sentences ("Let me think.",
       "Okay so.", "Hmm.") -- they must NOT
       appear in any criterion's evidence
       list.
3) Focus on quality assessment of the
     reasoning process, NOT only whether the
     final answer matches the GROUND TRUTH.
4) Consider ALL text in MODEL_OUTPUT
     including reasoning in <THINK> tags.
5) Output valid JSON only, no additional
     text.

CRITERIA (score range: {scores_info}):
[1] Problem Setup & Variable Identification:
      Does the response correctly identify
     variables, given conditions, constraints
     , and what is being asked?
[2] Concept/Theorem Selection: Does the
     response invoke the correct principle,
     theorem, formula, or domain knowledge
     for this problem?
[3] Logical Reasoning: Does each
     intermediate reasoning step logically
     follow from the prior steps and stated
     assumptions?
[4] Symbolic/Numeric Computation: Are the
     arithmetic, algebraic manipulation,
     equation solving, or numeric
     computations performed correctly?
[5] Final Answer Correctness: Does the final
      stated or boxed answer match the GROUND
      TRUTH ANSWER (allowing equivalent
     algebraic forms)?

OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
         score_info_identifier}, "sentences":
          [ "Exact sentence from MODEL_OUTPUT
         ", ... ]}},
    {{"id": 2, "score": {
         score_info_identifier}, "sentences":
          [ ... ]}},
    {{"id": 3, "score": {
         score_info_identifier}, "sentences":
          [ ... ]}},
    {{"id": 4, "score": {
         score_info_identifier}, "sentences":
          [ ... ]}},
    {{"id": 5, "score": {
         score_info_identifier}, "sentences":
          [ ... ]}}
  ],
  "explanation": "Optional brief explanation
       "
}}

CONTEXT
GROUND TRUTH ANSWER: {co_str}

PROBLEM:
{prompt}

MODEL_OUTPUT (to evaluate):
{response}
```

### G.3 Criteria and Grader Prompt for Medical

For the Medical corpus the task is *clinical multiple-choice reasoning* over MedQA-USMLE and MedMCQA items. As with STEM, answer correctness is decided by letter matching outside the grader; the prompt below states explicitly that the judge is *not* scoring the final answer, so the rubric contributes only the token-level reasoning term and the two signals stay disentangled.

- $c_1$: *Clinical Information Extraction* — Does the response correctly identify the salient clinical findings from the question (symptoms, signs, demographics, history, lab/imaging results)?
- $c_2$: *Differential Reasoning* — Does the response consider plausible alternative diagnoses or explanations rather than committing to one immediately?
- $c_3$: *Diagnostic/Therapeutic Justification* — Does the response link its reasoning to specific evidence (matching presenting findings to a condition's characteristic profile, or matching a treatment to its indication and mechanism)?
- $c_4$: *Medical Knowledge Accuracy* — Are the stated medical facts (pathophysiology, drug mechanism, diagnostic criteria, dosing, side effects, anatomy) factually correct?

**Prompt for grader model (GSSPO, Medical)**

```
You are an evaluator for a clinical
     reasoning task.

GOAL
Given a CLINICAL QUESTION, the GROUND TRUTH
     ANSWER (for your reference only), and a
     MODEL_OUTPUT (which may include <THINK
     >...</THINK> reasoning plus the final
     answer), evaluate the **quality of the
     reasoning process** in MODEL_OUTPUT.
```

```
You are NOT judging whether the final answer
     matches GROUND TRUTH -- a separate rule
     verifier handles that. Your job is to
    score the per-criterion reasoning
    quality and ground each non-zero score
    in specific sentences from the model's
    reasoning.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
2) For each criterion with score!=0, provide
     sentence-level evidence ({
    neg_score_note}):
   - Sentences must be verbatim from
       MODEL_OUTPUT only -- no paraphrasing.
   - Use "ALL" if the entire response
       uniformly satisfies the criterion (
       rare).
   - Provide multiple sentences if applicable
       .
   - Skip filler sentences ("Let me think.",
       "Hmm.", "Okay so.") -- they must NOT
       appear.
3) Consider ALL text in MODEL_OUTPUT
    including reasoning in <THINK> tags.
4) Output valid JSON only, no additional
    text.

CRITERIA (score range: {scores_info}) --
    reasoning process only:
[1] Clinical Information Extraction: Does
    the response correctly identify the
    salient clinical findings from the
    question (symptoms, signs, demographics,
     history, lab/imaging results)?
[2] Differential Reasoning: Does the
    response consider plausible alternative
    diagnoses or explanations rather than
    committing to one immediately?
[3] Diagnostic/Therapeutic Justification:
    Does the response link its reasoning to
    specific evidence (matching presenting
    findings to a condition's characteristic
     profile, or matching a treatment to its
     indication and mechanism)?
[4] Medical Knowledge Accuracy: Are the
    stated medical facts (pathophysiology,
    drug mechanism, diagnostic criteria,
    dosing, side effects, anatomy) factually
     correct?

OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
        score_info_identifier}, "sentences":
         [ "Exact sentence from MODEL_OUTPUT
        ", ... ]}},
    {{"id": 2, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 3, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 4, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}}
  ],
  "explanation": "Optional brief explanation
      "
}}

CONTEXT
GROUND TRUTH ANSWER (reference only -- do
    NOT use as a scoring criterion): {co_str
    }

CLINICAL QUESTION:
{prompt}

MODEL_OUTPUT (to evaluate):
{response}
```

### G.4 Criteria and Grader Prompt for Instruction Following

For the Instruction Following corpus the verifiable signal is not an answer key but a set of rule-based constraint checkers that test each instruction's formal requirements (length, format, forbidden tokens, and so on). Those checkers produce the correctness flag; the rubric below scores whether the response *reasons about* the constraints it is required to satisfy, and supplies the token-level term.

- $c_1$: *Constraint Decomposition* — Does the response correctly identify and parse every explicit constraint (e.g., enumerating "I need exactly 5 paragraphs separated by ***" / "must contain 'kaleidoscope' once" / "no commas allowed")? Implicit handling or generic acknowledgment without naming the specific rule does not count.
- $c_2$: *Strategy Articulation* — Does the response lay out a concrete satisfaction plan for each constraint before generation (e.g., "I'll target ∼80 words per paragraph", "I'll insert the keyword in sentence 2 of paragraph 3", "I'll avoid 'and' / 'but' to keep sentences short")? Generic phrases like "I will follow the rules" do not count; specific actionable tactics do.
- $c_3$: *Constraint-Aware Reasoning* — Is each generation step shaped by the constraints? Reward visible moments where the model revises an approach due to a constraint (e.g., "this clause needs a comma — I'll rewrite it as two sentences", "this draft has 4 bullets, I need to add one more before submitting").
- $c_4$: *Self-Verification* — Does the response perform explicit verification — counting para-

graphs/sentences/words, scanning for forbidden tokens, validating JSON brackets, re-reading for compliance — and self-correct after detecting a violation? Reward visible verify-and-fix loops.

**Prompt for grader model (GSSPO, Instruction Following)**

```
You are an evaluator for an instruction-
    following reasoning task.

GOAL
Given a USER INSTRUCTION (PROMPT) that
    carries explicit format/style/structural
     CONSTRAINTS, the CONSTRAINT_DESCRIPTION
     (for your reference only), and a
    MODEL_OUTPUT (which may include <THINK
    >...</THINK> reasoning plus the final
    answer), evaluate the **quality of the
    reasoning process** in MODEL_OUTPUT.

You are NOT judging whether the final answer
     mechanically satisfies the constraints
    -- a separate rule verifier handles that.
     Your job is to score the per-criterion
    reasoning quality and ground each non-
    zero score in specific sentences from
    the model's reasoning.

INSTRUCTIONS
1) For each criterion below, output {
    score_info_identifier}
2) For each criterion with score!=0, provide
     sentence-level evidence ({
    neg_score_note}):
   - Sentences must be verbatim from
      MODEL_OUTPUT only -- no paraphrasing.
   - Use "ALL" if the entire response
      uniformly satisfies the criterion (
      rare).
   - Provide multiple sentences if applicable
      .
   - Skip filler sentences ("Let me think.",
      "Hmm.", "Okay so.") -- they must NOT
      appear.
3) Consider ALL text in MODEL_OUTPUT
    including reasoning in <THINK> tags.
4) Output valid JSON only, no additional
    text.

CRITERIA (score range: {scores_info}) --
    reasoning process only:
[1] Constraint Decomposition: Does the
    response correctly identify and parse
    every explicit constraint (e.g.
    enumerating "I need exactly 5 paragraphs
     separated by ***" / "must contain '
    kaleidoscope' once" / "no commas allowed
    ")? Implicit handling or generic
    acknowledgment without naming the
    specific rule does not count.
[2] Strategy Articulation: Does the response
     lay out a concrete satisfaction plan
    for each constraint before generation (e.
    g. "I'll target ~80 words per paragraph
    ", "I'll insert the keyword in sentence
    2 of paragraph 3", "I'll avoid 'and' / '
    but' to keep sentences short")? Generic
    phrases like "I will follow the rules"
    do not count; specific actionable
    tactics do.
[3] Constraint-Aware Reasoning: Is each
    generation step shaped by the
    constraints? Reward visible moments
    where the model revises an approach due
    to a constraint (e.g. "this clause needs
     a comma -- I'll rewrite it as two
    sentences", "this draft has 4 bullets, I
     need to add one more before submitting
    ").
[4] Self-Verification: Does the response
    perform explicit verification --
    counting paragraphs/sentences/words,
    scanning for forbidden tokens,
    validating JSON brackets, re-reading for
     compliance -- and self-correct after
    detecting a violation? Reward visible
    verify-and-fix loops.

OUTPUT FORMAT (JSON only):
{{
  "criteria": [
    {{"id": 1, "score": {
        score_info_identifier}, "sentences":
         [ "Exact sentence from MODEL_OUTPUT
        ", ... ]}},
    {{"id": 2, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 3, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}},
    {{"id": 4, "score": {
        score_info_identifier}, "sentences":
         [ ... ]}}
  ],
  "explanation": "Optional brief explanation
      "
}}

CONTEXT
CONSTRAINT_DESCRIPTION (reference only -- do
     NOT use as a scoring criterion): {
    co_str}

PROMPT (user instruction):
{prompt}

MODEL_OUTPUT (to evaluate):
{response}
```

# H PACF Construction Details and Statistics

## H.1 Construction Pipeline

PACF consists of three stages. **Stage A: Conversation Preprocessing** removes malformed, low-signal, or erroneous turns (e.g., empty utterances), normalizes formatting across datasets, and compresses long dialogue histories into a *profile sketch*

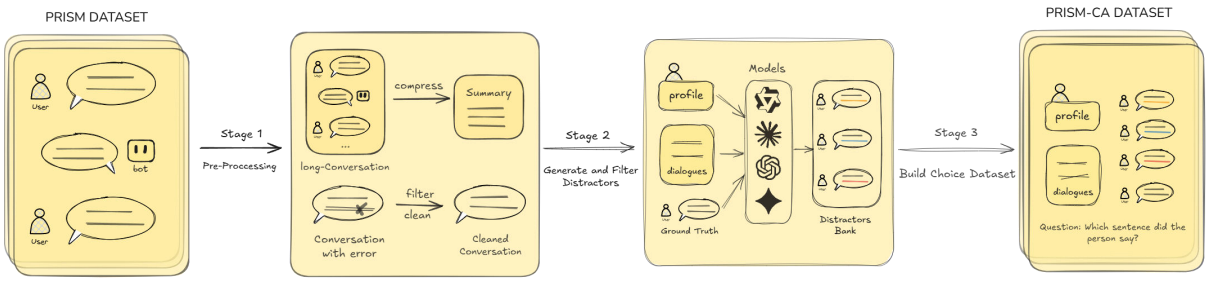



Figure 4: **PACF construction pipeline.** Stage A standardizes raw dialogues and compresses long histories into a profile sketch and minimal conversational context. Stage B generates candidate responses, fixes the original human-selected response as the correct option, and filters alternatives into a distractor bank. Stage C packages the profile, context, correct option, and distractors into a multiple-choice benchmark item.

and a *minimal context*. The profile sketch retains stable and preference-relevant user attributes, while the minimal context captures the current user query and preceding information needed for coreference resolution and task completeness.

**Stage B: Candidate Generation and Distractor Bank Construction** generates a diverse pool of candidate responses for each $(\text{profile}, \text{context})$ pair by prompting multiple large language models (OpenAI, 2024; DeepSeek-AI et al., 2025; Bai et al., 2023; Qwen et al., 2025; Yang et al., 2025; Gemini Team et al., 2025) with varied generation strategies and controlled stochasticity. The human-preferred response from the original dataset is fixed as the ground truth. Remaining candidates are filtered for safety, non-redundancy, contextual consistency, and profile plausibility without equivalence to the ground truth. A lightweight scorer estimates option alignment to select competitive distractors while keeping final labels grounded in the original human choices.

**Stage C: Multiple-Choice Packaging and Splitting** packages each item with a profile sketch, minimal conversational context, and 4–7 response options, including exactly one human-preferred answer and $K \in [3, 6]$ distractors sampled from the calibrated bank. Option order is randomized. We construct both in-domain persona splits and held-out-persona splits, enabling evaluation under matched persona distributions as well as out-of-domain persona generalization.

### H.2 Distractor Taxonomy and Dataset Statistics

PACF uses ten distractor types. Here, (w/) means the true output is provided to the generator model, and (w/o) means the true output is not provided.

- **Style violation:** Definitely incorrect with respect to the TARGET's intent (e.g., subtly contradict a key detail, flip a polarity, pick a wrong entity, omit a crucial constraint)
- **Topic violation:** Stay near the topic, but shifts focus to a closely related yet incorrect entity/attribute/option.
- **Richness violation:** If TARGET is detailed, produce a concise variant that omits a crucial condition so it becomes wrong; if TARGET is brief, produce a more elaborate variant that adds an incorrect detail.
- **Free violation:** Convey a different, clearly incorrect intention/fact compared with the TARGET.
- **Profile violation(w/):** Keep the same language and sentence type as TARGET, but clearly contradicts the PROFILE (opposite trait/preference/stance).
- **Conversation violation(w/):** Keep the same language and sentence type as TARGET, but subtly disregards the conversation requirement (e.g., answers a related but different question, ignores a key constraint, wrong perspective/recipient).
- **Both violation(w/):** Combination of Profile violation and Conversation violation.
- **Profile violation(w/o):** Same as Profile violation(w/) without true output.
- **Conversation violation(w/o):** Same as Conversation violation(w/) without true output.
- **Both violation(w/o):** Same as Both violation(w/) without true output.

We compute the perplexity of each distractor. To keep item difficulty balanced, distractors whose perplexities fall below a threshold are enhanced by reasoning models (Gemini2.5-Flash, o3-mini, o4-mini, gpt-5, gpt-5-mini).

Table 13 reports the distractor-type statistics for PRISM-CA and Big5Chat-CA. Table 14 reports the length distributions of profiles, conversations, and generated options.

# I Qualitative Analysis: Reward Granularity and Logic Attribution

In this section, we illustrate the core distinction between **GSSPO** and traditional Uniform Reward Assignment methods. Standard alignment approaches typically assign a scalar reward to the entire generated sequence. This coarse-grained feedback can be limiting for reasoning-intensive tasks because it does not distinguish between critical logical steps and neutral filler tokens. A sequence leading to a wrong answer may contain correct intermediate steps, and conversely, a correct answer can sometimes stem from flawed logic or spurious correlation.

**GSSPO** addresses this by supplementing *outcome correctness* with *process-level validity*. As illustrated in Figure 5, our method performs dense, span-level attribution to provide feedback not only on *whether* an answer is right or wrong, but also on *where* the reasoning appears valid or invalid.

## I.1 Case Study: Locating Validity and Hallucination

Figure 5 presents a side-by-side comparison of how the GSSPO Judge evaluates reasoning trajectories.

**(a) Positive Attribution (The "Why" of Correctness):** In the positive case (Figure 5a), the model correctly identifies the target persona ("Low Openness") and derives the associated behavior ("Stability/Routine"). Unlike uniform methods that would simply reward the final token "Option D", GSSPO explicitly assigns positive value ($+V_{sr}$) to the intermediate reasoning spans (highlighted in cyan). This reinforces the specific logical bridge: *Trait* $\rightarrow$ *Behavioral Constraint* $\rightarrow$ *Option Selection*. By rewarding the reasoning process ("A low openness person might prefer..."), the feedback encourages the model to connect the selected option to the persona profile rather than relying only on the final answer token.

**(b) Negative Attribution (The "Where" of Error):** The negative case (Figure 5b) illustrates how GSSPO differs from uniform penalties. Here, the model commits a **Subject Misalignment** error. The task requires the agent (Stefon) to respond to a user (Rashaan), but the model mistakenly attributes the user's trait (Agreeableness) to the agent's persona. A uniform penalty would punish the entire chain, potentially suppressing the model's general ability to reason about "agreeableness." In contrast, GSSPO localizes the penalty ($-V_{sr}$) specifically to the hallucinated premise ("So Rashaan is agreeable... meaning he's likely to be positive"). This feedback distinguishes between understanding the trait and applying it to the correct participant. Such fine-grained feedback can help the model distinguish between a wrong fact and a wrong logical application.

Table 13: Distribution of distractor types comparison between PRISM-CA and Big5Chat-CA.

| | | PRISM-CA | | | Big5Chat-CA | | |
|---|---|---|---|---|---|---|---|
| **Category** | **Types** | **Amt.** | **Ratio** | **Grp.** | **Amt.** | **Ratio** | **Grp.** |
| **Basic Types** | Style violation | 7218 | 6.86 | 51.9 | 1153 | 6.97 | 60.1 |
| | Topic violation | 7268 | 6.90 | | 1397 | 8.45 | |
| | Richness violation | 8333 | 7.92 | | 1340 | 8.10 | |
| | Free violation | 6807 | 6.47 | | 1097 | 6.63 | |
| | Profile vio. (w/) | 5458 | 5.18 | | 929 | 5.62 | |
| | Conv. vio. (w/) | 6473 | 6.15 | | 1045 | 6.32 | |
| | Both vio. (w/) | 5469 | 5.20 | | 897 | 5.42 | |
| | Profile vio. (w/o) | 796 | 0.76 | | 586 | 3.54 | |
| | Conv. vio. (w/o) | 3934 | 3.74 | | 725 | 4.38 | |
| | Both vio. (w/o) | 2886 | 2.74 | | 769 | 4.65 | |
| Correct output | | 19161 | 18.20 | | 2996 | 18.12 | |
| Enhanced distractor | | 31475 | 29.90 | | 3604 | 21.79 | |
| **Total** | | **105278** | **100** | | **16538** | **100** | |

Table 14: Distribution of lengths for Profiles, Conversations, and Generated Outputs across datasets.

| **Category** | **Quantile** | **PRISM-CA** | **Big5Chat-CA** |
|---|---|---|---|
| **Profile** | Min | 196 | 198 |
| | Q1 | 352 | 317.8 |
| | Median | 417 | 353 |
| | Q3 | 512 | 397 |
| | Max | 1748 | 561 |
| **Conversation** | Min | 51 | 43 |
| | Q1 | 438 | 188 |
| | Median | 868 | 213 |
| | Q3 | 1637 | 238 |
| | Max | 7616 | 309 |
| **Distractors & Outputs** (Generated) | Min | 0 | 4 |
| | Q1 | 49 | 136 |
| | Median | 76 | 201 |
| | Q3 | 115 | 242 |
| | Max | 2151 | 1082 |

**(a) Positive Case: Rewarding Persona-Aligned Reasoning**

**Target Profile (Tristen):**
*Trait:* Low Openness. *Goal:* Select stability/routine.

**Model Reasoning Trace (<think>):**
`...I need to choose the response that aligns with Tristen's low openness.`
**A low openness person might prefer stability, routine, and less change.**
`...Option A says ... big changes... [Skips]...`
**Option D ... aligns with a low openness trait because they avoid big changes and prefer stability.**
`So the correct answer should be D...`

**Judge Attribution (Reward $+V_{sr}$):**

- **A low openness person...**
  ↪ **[1] Persona Consistency**: Correctly interprets trait.
- **Option D ... aligns with...**
  ↪ **[5] Affective Alignment**: Maps constraint to option.

**(b) Negative Case: Penalizing Logic Failure**

**Task Input:**
*Profile:* PersonX (Rashaan) = Agreeable.
*Goal:* Select response **for Responder (Stefon)**.

**Model Reasoning Trace (<think>):**
`...The trait to show is agreeableness (Rashaan)...`
**So Rashaan is agreeable... meaning he's likely to be positive...**
`...Option B: "Oh, Rashaan..."`
**Option B ... aligns with Rashaan being agreeable ... So the best choice is B.**

**Judge Attribution (Penalty $-V_{sr}$):**

- **So Rashaan is agreeable...**
  ↪ **[4] Contextual Grounding**: **Subject Misalignment**. Applies trait to user instead of agent.
- **...aligns with Rashaan...**
  ↪ **[2] Conversational Coherence**: **Invalid Inference**. Justification based on wrong role.

Figure 5: **Qualitative Analysis of GSSPO Judge Mechanism. (a)** The model receives positive rewards ($+V_{sr}$, blue) for correctly defining the persona and mapping it to the target option. **(b)** The model is penalized ($-V_{sr}$, red) for hallucinating the subject of the personality trait (confusing the user Rashaan with the agent Stefon), even if it accidentally selects the correct option.

## J Preliminary

### J.1 GRPO

GRPO can be viewed as an adaptation of PPO (Schulman et al., 2017) for LLM training. In PPO, a value model is used as a baseline for the advantage function to reduce variance. For LLM generation, however, rewards are often observed only at the sequence level, making token-level value estimation difficult.

GRPO (Shao et al., 2024) uses the mean reward of multiple sampled outputs as the baseline, dispensing with the value model entirely. We sample outputs $o_1, \ldots, o_G$ from the old policy $\pi_{\theta_{old}}$, write the token-level importance ratio as

$$w_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})},$$

and maximize

$$J_{GRPO}(\theta) = \mathbb{E}_{q\sim P(Q),\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big\{\min\Big[w_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\Big] - \beta D_{KL}[\pi_\theta||\pi_{ref}]\Big\},$$

where $\epsilon$ and $\beta$ are hyperparameters, and $\hat{A}_{i,t}$ is the advantage estimated from the relative reward of each output within the group.

Following prior PPO variants, we use an unbiased estimator of the KL divergence:

$$D_{KL}[\pi_\theta||\pi_{ref}] = \frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - 1$$

which is non-negative.

### J.2 GSPO

PPO and GRPO can face instability and efficiency issues on large models. GSPO (Zheng et al., 2025) addresses these issues by shifting the importance ratio from the token level to the sequence level.

To describe GSPO, we first revisit GRPO. In GRPO, a group of outputs is generated for each question, and their relative rewards are used for optimization. The objective applies a token-level importance ratio to each generated token and combines it with the group-relative advantage.

This token-level ratio can become noisy for long sequences, increasing the risk of unstable gradients. GSPO instead starts from the sequence level: reward computation, clipping, and importance weighting are applied to the whole output. In this view, the key idea of GSPO is to align the granularity of rewards, updates, and importance ratios.

Now we give the sequence-level GSPO optimization objective:

$$J_{GSPO}(\theta) = \mathbb{E}_{x\sim D,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(\sim|x)} \left[\frac{1}{G}\sum_{i=1}^{G}\min\left(s_i(\theta)\hat{A}_i, \text{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right)\right]$$

where we adopt the group-based advantage estimation:

$$\hat{A}_i = \frac{r(x, y_i) - mean(\{r(x, y_i)\}_{i=1}^{G})}{std(\{r(x, y_i)\}_{i=1}^{G})},$$

and define the importance ratio $s_i(\theta)$ based on sequence likelihood:

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{old}}(y_i|x)}\right)^{\frac{1}{|y_i|}} = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{old}}(y_{i,t}|x, y_{i,<t})}\right).$$

Actually we can also put the whole structure into token-level. We only need to change our $s_i(\theta)$ into

$$s_{i,t}(\theta) = sg[s_i(\theta)] \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{sg[\pi_\theta(y_{i,t}|x, y_{i,<t})]}$$

Here $sg[]$ denotes only taking the numerical value but stopping the gradient.

## K Algorithm Pseudocode

Algorithm 1 provides the complete pseudocode for GSSPO training procedure. This captures the core ideas of inter-step linear decay (the $\alpha$ value) and rubric-based reward computation (using the LLM judge).

**Default reward closed forms.** $S_{\text{def}}$ selects the default-token-reward mode and is set to `"zero"` in every run reported in this paper. Under this setting the two default rewards invoked in Algorithm 1 are

$$R_d^{\text{resp}} = \mathbf{1}[\text{is_correct}], \qquad R_d^{\text{token}} = \mathbf{0}, \quad (1)$$

so the algorithm's $\alpha$=0 clamp on $R^{\text{token}}$ is consistent with, rather than additional to, the default mode. Consequently the free-exploration phase ($\alpha$=0) carries no token-level term at all and the objective reduces term-by-term to GSPO; the GSPO row of Table 2 is the empirical anchor for that limit.

**Algorithm 1** GSSPO Algorithm

**Input:** Policy model $\pi_\theta$, data batch $\mathcal{D}$ with (prompts $q$, responses $o$), grader model $J_\phi$, global step $t$.
**Initialize:** Guidance end step $T_{\text{end}}$, blend span $T_{\text{blend}}$, default strategy $S_{\text{def}}$, criterion weight vector $\mathbf{r}$.
**for** each training iteration at global step $t$ **do**
    ▷ Compute scaffolding decay ratio
    **if** $t < T_{\text{end}}$ **then**
        $\alpha \leftarrow 1.0$ ▷ Full guidance phase
    **else if** $t < T_{\text{end}} + T_{\text{blend}}$ **then**
        $\alpha \leftarrow 1.0 - (t - T_{\text{end}})/T_{\text{blend}}$ ▷ Linear blending phase
    **else**
        $\alpha \leftarrow 0.0$ ▷ Default/No guidance phase
    **end if**
    Let $\mathcal{D}_{\text{judge}} \leftarrow \emptyset$
    **for** each sample $(q, o)$ in data batch $\mathcal{D}$ **do**
        Determine ‘is_correct‘ by comparing $o$ to ground truth.
        **if** $\alpha_t > 0$ and sample meets ‘judge_score_target‘ filter **then**
            Add $(q, o, \text{is_correct})$ to $\mathcal{D}_{\text{judge}}$
        **end if**
    **end for**
    ▷ Evaluate a batch with the grader, corresponds to $\mathbf{b}_i = J_\phi(q, o_i, \mathcal{C})$
    **if** $\mathcal{D}_{\text{judge}}$ is not empty **then**
        $B \leftarrow J_\phi(\mathcal{D}_{\text{judge}})$ ▷ Get rubric evaluations (scores and sentences)
    **end if**
    **for** each sample $(q, o)$ in data batch $\mathcal{D}$ **do**
        ▷ Compute the scalar quality score $s = \mathbf{b} \cdot \mathbf{r}$ and resulting rewards
        $R_g^{\text{token}}, R_g^{\text{resp}} \leftarrow \mathbf{0}, 0$ ▷ Initialize guided rewards to zero
        **if** sample was evaluated in $B$ **then**
            Let $\mathbf{b}$ be the rubric score vector from $B$ for this sample.
            $s \leftarrow \mathbf{b} \cdot \mathbf{r}$ ▷ Compute weighted quality score
            $R_g^{\text{token}} \leftarrow \text{TextToTokenReward}(\mathbf{b}, \text{is_correct})$
            $R_g^{\text{resp}} \leftarrow (\text{is_correct}?1.0 : 0.0) + s$
        **end if**
        $R_d^{\text{token}} \leftarrow \text{ComputeDefaultTokenReward}(S_{\text{def}}, \text{is_correct})$
        $R_d^{\text{resp}} \leftarrow \text{ComputeDefaultResponseReward}(\text{is_correct})$
        ▷ Blend rewards based on the linear decay ratio $\alpha$
        $R^{\text{token}} \leftarrow \alpha \cdot R_g^{\text{token}} + (1 - \alpha) \cdot R_d^{\text{token}}$
        $R^{\text{resp}} \leftarrow \alpha \cdot R_g^{\text{resp}} + (1 - \alpha) \cdot R_d^{\text{resp}}$
        **if** $\alpha = 0.0$ **then** ▷ Safety check for pure default phase
            $R^{\text{token}} \leftarrow \mathbf{0}$
        **end if**
        Store $R^{\text{token}}$ and $R^{\text{resp}}$ for policy update.
    **end for**
**end for**
**Return:** Computed token and response rewards for the batch.