VSPO: Vector-Steered Policy Optimization for Behavioral Control

Xuechen Zhang*
University of Michigan
zxuechen@umich.edu

Zijian Huang*
University of Michigan
zijianh@umich.edu

Kai Yang
University of Michigan
yangkai@umich.edu

Weijia Zhang
University of Michigan
wjzhang@umich.edu

Jiasi Chen
University of Michigan
jiasi@umich.edu

Samet Oymak
University of Michigan
oymak@umich.edu

Abstract

Modern language models often need to optimize a primary accuracy objective while also accommodating secondary behavioral preferences, such as verbosity, agreeableness, or the level of technical expertise in its response. In practice, a base model may exhibit a desired behavior very rarely or not at all. Thus, endowing the model with a target behavior creates a *sparse behavioral reward* bottleneck. To address such multi-objective problems, we introduce Vector-Steered Policy Optimization (VSPO) which employs a steering vector associated with the target behavior to control the behavior intensity of the generated rollouts. VSPO is obtained by modifying GRPO to sample rollouts with varying steering intensities. This process can be interpreted as latent self-distillation from the model's own steered rollouts, through which the model internalizes its steering vector. By varying steering intensities, VSPO *upsamples* rare behaviors and enriches rollout diversity, which alleviates the sparse reward issue and provably accelerates the policy optimization. Through comprehensive theory and experiments, we establish that VSPO has favorable properties compared to vanilla reward shaping and other alternative approaches. Specifically, under a bandit abstraction, VSPO provably achieves better iteration complexity than reward-shaped GRPO when the steering-induced distributions are sufficiently aligned with the target behavior. We evaluate VSPO across multiple reasoning benchmarks, including MATH and MMLU-Pro, for four target behaviors: explanation expertise, confidence expression, robustness to misleading context, and response verbosity. Our results show that VSPO consistently improves the control along target behavior while maintaining or improving task accuracy compared with reward shaping, teacher-trace distillation, and guidance-based baselines.

1 Introduction

Recent years witnessed remarkable progress on post-training and reasoning capabilities of large language models (LLM) [17, 19]. These models can now tackle highly complex mathematical and scientific problems. However, many desirable model behaviors are not captured by correctness alone. In reasoning tasks, we often want the model to maintain a primary objective like answer correctness, while also controlling secondary characteristics such as verbosity, confidence expression, expertise level, or agreeableness. This setting is closely related to multi-objective alignment, where models must trade off several human-preference dimensions rather than optimizing a single scalar notion

*These authors contributed equally to this work. Correspondance to {zxuechen,zijianh,oymak}@umich.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

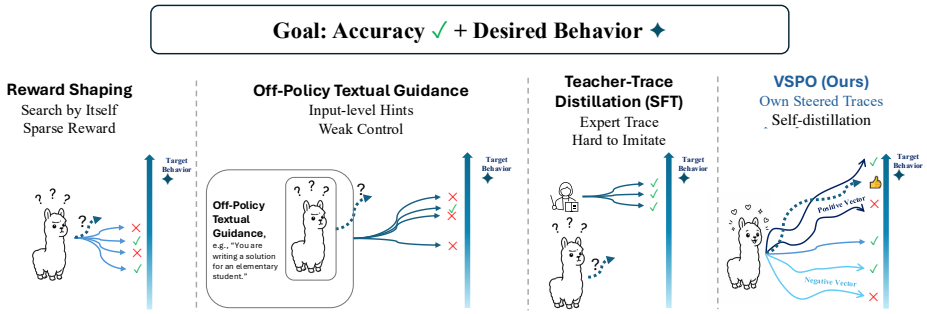


Figure 1: **Motivation and overview of VSPO.** The goal is to improve task accuracy while inducing a desired reasoning behavior. The figure illustrates VSPO and alternative methods for the goal of improving both task accuracy and a target behavior. Check marks and crosses indicate whether each generated trace is correct or incorrect. The vertical blue arrow denotes the target-behavior direction: traces higher along the arrow better exhibit the desired behavior. VSPO uses latent steering vectors to generate a structured family of the model's own steered rollouts, spanning different behavior intensities while remaining close to the current policy. By rewarding accurate and behavior-aligned traces and distilling them back into the unsteered policy, VSPO enables latent self-distillation from its own steered rollouts toward both correctness and controllable reasoning behavior without requiring inference-time textual guidance. The bold dashed trajectory refers to the rollout generated at inference-time and thumbs-up indicates that, after VSPO training, the model can generate correct target-behavior traces.

of quality. Prior work has explored multi-objective or dynamically adjustable reward formulations [26, 28], but such methods still primarily operate at the reward level and inherit the usual difficulties of reward design, scaling, and optimization stability.

A straightforward approach is to apply GRPO with an auxiliary shaped reward for the desired behavior [1, 23, 28, 32]. While intuitive and simple, reward shaping can be inefficient when the target behavior is rarely expressed in the sampled rollouts. Scalar rewards only evaluate generated trajectories; they do not directly improve the diversity or quality of the rollout distribution. If the majority of responses in a GRPO group are low-quality or behaviorally similar, relative advantage normalization provides little useful signal for learning the desired characteristic. Another common approach is to use a stronger teacher model to generate or rewrite reasoning traces endowed with the target behavior and then fine-tune the student model on these traces. Teacher traces can provide cleaner demonstrations of the desired behavior, but they are off-policy with respect to the student: the student is trained on trajectories produced by another model rather than on its own sampled outputs. This off-policy mismatch is one of the motivations behind recent on-policy self-distillation methods [21, 10, 33]. A third approach is to use powerful teacher models during online RL, for example as AI-feedback annotators, judges, or rewriters to achieve the target behavior. RLAIF [7, 15] replaces or augments human feedback with feedback from AI models, and online AI-feedback methods query an LLM annotator during training to obtain preference labels for current-policy samples. These methods improve scalability relative to human annotation and can provide controllable feedback, but repeated online calls to powerful API models are expensive, especially when used for multi-rollout generation or trace rewriting inside the RL loop. Moreover, training the student on rewritten traces still incur off-policy drawbacks [20].

As depicted in Figure 1, we propose Vector-Steered Policy Optimization (VSPO) to address these bottlenecks. Recent work on steering vectors [5] shows that behavioral traits can be represented as latent directions in a model's activation space, suggesting a way to quantitatively characterize and steer target behaviors beyond surface-level prompting or scalar rewards. VSPO relies on a steering vector associated with the target behavior. In practice, we construct this vector using a more powerful teacher model to obtain contrastive rewrite pairs associated with the behavior. During RL, the current policy is rolled out under multiple steering intensities, producing a diverse and controllable rollout distribution. Because these rollouts are still generated by the target model, training remains anchored to the current policy. We then apply GRPO with a multi-objective reward, where the behavioral reward is proportional to the steering intensity. The full procedure is depicted in Figure 2.

Overall, our paper makes the following contributions:

- We introduce VSPO, a multi-objective policy optimization method that leverages steering vectors to control language model behavior while preserving task accuracy. Compared to vanilla GRPO or reward shaping, VSPO actively improves rollout diversity by explicitly sampling traces along the target latent vector. Compared with SFT on teacher traces, it avoids directly training on off-policy teacher trajectories: the teacher can assist vector construction, but the RL process is fully guided by the student rollouts. Compared with RLAIF or online rewriting, VSPO substantially reduces reliance on repeated teacher calls, since the steering vector is reused throughout optimization.
- We formally characterize the advantage of VSPO over reward shaping: We establish how VSPO can provably achieve better iteration complexity than reward-shaped GRPO under a bandit abstraction by upsampling sparse but desirable behavioral patterns (see Section 3.2, Propositions 1, 2).
- We empirically evaluate VSPO on MATH and MMLU-Pro across four target behavior types (see Section 4.1.2 for details) and show that it consistently outperforms reward shaping, teacher-trace SFT, and RLAIF-style baselines while maintaining correctness.
- VSPO has favorable properties: Compared to alternatives, it incurs mild distributional shift from base model (see Fig. 6). It induces a natural learning curriculum along the behavior axis throughout policy optimization (see Fig. 7). The behavioral component of its reward function is normalized by design as it corresponds to *one unit of steering vector*.

2 Related Work

Teacher-trace distillation is a common technique to guide target models, where a stronger model generates reasoning traces and the target model is trained with supervised fine-tuning. This paradigm has been widely used for improving reasoning and controlling response format [8, 13, 16]. However, teacher-trace SFT is off-policy: the student learns from trajectories generated by an external model rather than from its own sampled rollouts, which can cause a mismatch between the distilled behavior and the student policy's reachable distribution [6, 32]. RL post-training methods, especially RLVR, address this by optimizing the model on its own sampled responses using verifiable rewards and group-relative advantages [19, 6]. Yet standard GRPO relies on ordinary rollout sampling and scalar rewards, which may be insufficient when the goal is to preserve correctness while controlling secondary characteristics such as conciseness, confidence expression, expertise level, or sycophancy resistance. Another line of work incorporates target characteristics directly into rewards, such as length penalties or LLM-based preference scores, and more broadly studies multi-objective alignment [23, 1, 32]. However, reward shaping only changes scalar feedback after rollout generation and does not directly improve the diversity of the sampled candidates. In contrast, VSPO targets controllable secondary behaviors by shaping the rollout distribution through latent steering vectors. Similarly, textual guidance and RLAIF-style methods use auxiliary instructions, critiques, or AI feedback to elicit desired behaviors during RL [22, 30, 27, 29], but repeated external guidance can be expensive and may introduce mismatch between guided training and unguided inference. Recent self-distillation methods reduce off-policy mismatch by deriving learning signals from the model itself. SDFT uses a demonstration-conditioned version of the model as a self-teacher, while SDPO uses high-quality self-generated rollouts as implicit optimization targets [21, 10]. Related work also improves RLVR through refined credit assignment, denser intermediate signals, and multi-attempt verifier feedback [14, 11]. These methods are related to our motivation, but mainly target task-performance rather than controllable secondary behavior. Finally, recent curriculum-based RL methods have been shown to help overcome optimization bottlenecks in SFT and RL training [31]. VSPO induces a natural curriculum along the behavior axis by varying steering intensity, and we empirically show that this improves over naive applications of SFT and RL for acquiring target behaviors.

Our work is also related to representation engineering and vector steering, which identify semantically meaningful directions in activation space and manipulate model behavior at inference time [34, 24, 18]. Persona vectors further specialize such directions to behavioral traits [5]. Unlike prior work that primarily uses steering as a test-time intervention, VSPO uses vector steering as a mechanism for latent self-distillation from steered model generations. The details of related works is at Appendix C.

3 Proposed Method: Vector-Steered Policy Optimization (VSPO)

3.1 Algorithm Overview

The workflow of VSPO is shown in Figure 2 and proceeds as follows:

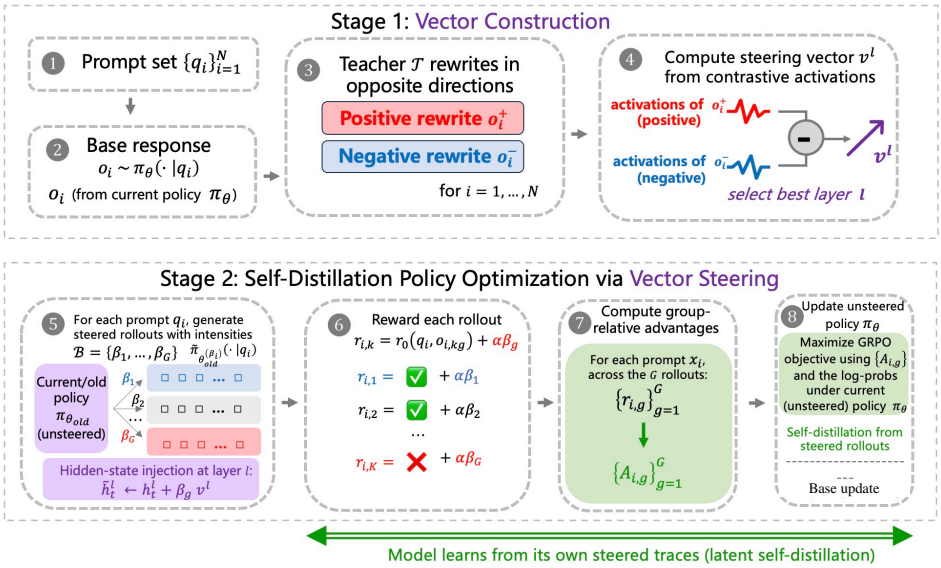


Figure 2: **Overview of VSPO algorithm**. In Stage 1, prompts are sampled from the current policy, a teacher rewrites the responses into contrastive positive and negative directions, and their activation differences are used to construct a steering vector. In Stage 2, the current policy generates on-steered-policy rollouts under different steering intensities, receives rewards summing task correctness and steering-dependent preference signals, and is updated with GRPO. The final update is applied to the unsteered policy, enabling the model to distill useful behaviors from its own steered traces.

- **Steering Vector Construction**: In the first stage, the target model generates base responses for a small set of sampled questions. A teacher model then rewrites each response into positive and negative behavioral variants, such as expert versus elementary explanations for the expertise-level axis. Using the original prompt, we extract target-model activations for both variants, compute their mean activation difference at each layer, and select the strongest layer-wise direction as the steering vector.
- **Self-Distillation Policy Optimization**: In the second stage, for each sampled prompt q_i,
  1. **Rollout with intensities**: Sample a group of G rollouts $\{o_{i,g}\}_{g=1}^G$ with intensities $\mathcal{B} = \{\beta_g\}_{g=1}^G$.
  2. **Reward computation**: For each rollout $o_{i,g}$, the reward $r_{i,g}$ is a combination of the primary reward and the proxy behavioral reward, $r_0(q_i, o_{i,g}) + \alpha\beta_g$, where r_0 is the primary reward function, and α is a steering preference hyperparameter.
  3. **Policy updates**: Optimize the policy via the following objective:

$$\mathcal{L}_{VSPO}(\theta) = \mathbb{E}_x \left[\frac{1}{G} \sum_{g=1}^{G} \frac{1}{T_g} \sum_{t=1}^{T_g} \min\big(\rho_{g,t}(\theta) A_g,\ \text{clip}\big(\rho_{g,t}(\theta), 1-\epsilon, 1+\epsilon\big) A_g\big) \right] - \lambda \mathcal{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}). \quad (1)$$

where $\bar{r}, \sigma_r$ are the mean and standard deviation of the rewards in the same group

$$\rho_{g,t}(\theta) = \frac{\pi_\theta(o_{g,t} \mid q, o_{g,<t})}{\pi_{\theta_{\text{old}}}(o_{g,t} \mid q, o_{g,<t})}, \quad A_g = \frac{r_g - \bar{r}}{\sigma_r}. \quad (2)$$

The details of the algorithm can be found in Algorithm 1 and Appendix D.

3.2 Theoretical Analysis

VSPO can be viewed as a form of *distribution shaping* during sampling: Instead of drawing a group entirely from the current policy, it draws from multiple related but contrasted distributions. In this section, we quantify the benefit of distribution shaping obtained by vector steering, comparing to reward shaping, which is a typical approach to solving multiple-objective optimization in GRPO. We consider a trajectory-level bandit model in which each arm represents a complete response for a fixed prompt. This abstraction isolates the effect of steering on sampling and population policy updates.

Theoretical Setup. We consider a K-armed bandit with deterministic rewards, i.e., a one-step decision problem with K possible actions. Each arm $i \in [K]$ has a primary reward $x(i)$ and behavioral reward $y(i)$. The target arm $i^\star$ is defined as $i^\star = \arg\max_{i \in [K]} \{x(i) : y(i) = \max_{j \in [K]} y(j)\}$.

Denote π_t as policy at iteration t. Define scalarized reward $r(i) := x(i) + \alpha y(i)$, and expected reward of policy π as $J(\pi) := \sum_{i=1}^{K} \pi(i) r(i)$. We assume $\pi_0(i) > 0$ for all $i \in [K]$, and $\alpha > \max_{i:\, y(i)<y(i^\star)} \frac{\max\{x(i)-x(i^\star),0\}}{y(i^\star)-y(i)}$ so that $i^\star$ is the unique maximizer of $r(i)$ (Proof in Appendix E.1).

For each suboptimal arm $i \neq i^\star$, define the scalar reward gap $\Delta_i := r(i^\star) - r(i)$, together with $\Delta_{\min} := \min_{i \neq i^\star} \Delta_i$, $\Delta_{\max} := \max_{i \neq i^\star} \Delta_i$. We also define the *bandit conditioning constant* λ as

$$\lambda := \frac{D_x + \alpha D_y}{\Delta_{\max}}, \quad \text{where } D_x := \max_{i\in[K]} x(i) - \min_{i\in[K]} x(i),\ D_y := \max_{i\in[K]} y(i) - \min_{i\in[K]} y(i). \tag{3}$$

GRPO. At iteration t, we draw G i.i.d. samples from the current policy π_t. The reward for each sample i is given by reward shaping $r(i) = x(i) + \alpha y(i)$. Let $\bar{r}_t^{\text{GRPO}}$ and $\bar{\sigma}_t^{\text{GRPO}}$ be the mean and standard deviation of the sampled scalar rewards. The empirical arm-level GRPO score is

$$\widehat{A}_t^{\text{GRPO}}(i) := N_t(i) \frac{r_i - \bar{r}_t^{\text{GRPO}}}{\bar{\sigma}_t^{\text{GRPO}}}, \tag{4}$$

where $N_t(i)$ is the number of times arm i appears in the group.

VSPO. At each iteration t, we assume applying vector steering returns two distributions π_t^+ and π_t^- and define $\bar{\pi}_t = \frac{\pi_t^+ + \pi_t^-}{2}$. VSPO draws $G/2$ samples from π_t^+ and $G/2$ samples from π_t^-. The shaped reward for samples from π_t^+, π_t^- are defined as r_t^+, r_t^- respectively:

$$r_t^+(i) := x(i) + \alpha \mu_{y,t}^+,\ r_t^-(i) := x(i) + \alpha \mu_{y,t}^-, \text{ where } \mu_{y,t}^+ := \mathbb{E}_{j\sim\pi_t^+}[y(j)],\ \mu_{y,t}^- := \mathbb{E}_{j\sim\pi_t^-}[y(j)].$$

Let $\bar{r}_t^{\text{VSPO}}, \bar{\sigma}_t^{\text{VSPO}}$ be the mean and standard deviation of all shaped rewards, and $N_t^+(i), N_t^-(i)$ be the number of times arm i appears in the groups π_t^+, π_t^-. The empirical arm-level VSPO score is

$$\widehat{A}_t^{\text{VSPO}}(i) := N_t^+(i) \frac{r_t^+(i) - \bar{r}_t^{\text{VSPO}}}{\bar{\sigma}_t^{\text{VSPO}}} + N_t^-(i) \frac{r_t^-(i) - \bar{r}_t^{\text{VSPO}}}{\bar{\sigma}_t^{\text{VSPO}}}. \tag{5}$$

KL-Regularized Update. Given an arm-level score vector $A_t \in \mathbb{R}^K$, we update the policy by

$$\pi_{t+1} = \arg\max_{\pi} \left\{ \frac{1}{G} \sum_{i=1}^{K} \frac{\pi(i)}{\pi_t(i)} A_t(i) - \frac{1}{\eta} \sum_{i=1}^{K} \pi(i) \log \frac{\pi(i)}{\pi_t(i)} \right\}. \tag{6}$$

In the theoretical analysis below, we use the population scores A_t^{GRPO} and A_t^{VSPO}, obtained by replacing the empirical quantities in Eqns. (4) and (5) with their population counterparts. The explicit population-score formulas and derivations are provided in Appendices E.2 and E.3.

Proposition 1 (Iteration complexity of GRPO). *Under the GRPO population score A_t^{GRPO} in Appendix E.2, we have $J(\pi_T) \geq r^\star - \varepsilon$ when*

$$T \geq T^{\text{GRPO}}(\varepsilon) := \frac{\Delta_{\max}}{2\eta \left(1 - \frac{1}{G}\right) \Delta_{\min}} \log \frac{C_0}{\varepsilon}. \tag{7}$$

Proposition 2 (Iteration complexity of VSPO). *Define distributional reward gaps*

$$\delta_{x,t} := \mathbb{E}_{i\sim\pi_t^+}[x(i)] - \mathbb{E}_{i\sim\pi_t^-}[x(i)],\ \delta_{y,t} := \mathbb{E}_{i\sim\pi_t^+}[y(i)] - \mathbb{E}_{i\sim\pi_t^-}[y(i)],\ \delta_t := \delta_{x,t} + \alpha\delta_{y,t},$$

distributional gap $d_t(i) := \frac{\pi_t^+(i) - \pi_t^-(i)}{2}$, and relative distributional contrast $\rho_t(i) := \frac{d_t(i)}{\pi_t(i)}$. Assume $\bar{\pi}_t = \pi_t$ for all $t \geq 0$, and the vector steering is γ-good: for all $t \geq 0$ and all $i \neq i^\star$,

$$\frac{\delta_t}{2G} \left(\rho_t(i^\star) - \rho_t(i)\right) \geq \gamma,\ \delta_{y,t} \left(\rho_t(i^\star) - \rho_t(i)\right) \geq 2 \left(y(i^\star) - y(i)\right). \tag{8}$$

Then, under the VSPO population score A_t^{VSPO} in Appendix E.3, we have $J(\pi_T) \geq r(i^\star) - \varepsilon$ when

$$T \geq T^{\text{VSPO}}(\varepsilon) := \frac{\lambda \Delta_{\max}}{2\eta \sqrt{1 - \frac{1}{G}} \left(\left(1 - \frac{1}{G}\right) \Delta_{\min} + \gamma\right)} \log \frac{C_0}{\varepsilon}, \tag{9}$$

where λ is the bandit conditioning constant defined in Eqn. (3).

Corollary 3. *When γ defined in Eqn. (8) satisfies $\gamma > \min\left\{\lambda, \frac{\lambda^2}{4}\right\}\Delta_{\min}$, we have $T^{\mathrm{VSPO}}(\varepsilon) < T^{\mathrm{GRPO}}(\varepsilon)$, where $T^{\mathrm{GRPO}}(\varepsilon), T^{\mathrm{VSPO}}(\varepsilon)$ are defined in Eqns. (7) and (9) respectively.*

The γ-good condition defined in Eqn. (8) has a direct interpretation. The first inequality requires the positive distribution π_t^+ to place relatively more mass on the target arm than on any suboptimal arm, producing an additional margin γ in the scalarized reward direction. The second inequality requires this distributional contrast to be aligned with the behavioral reward y. Thus, by Corollary 3 we know VSPO improves over GRPO (with reward shaping) when the positive and negative sampling distributions create a useful contrast between high-quality and low-quality regions of the response space.

On the other hand, we can show an upper bound for γ defined in Eqn. (8) (See Appendix E.8). Thus, in this setting, one cannot expect VSPO to achieve arbitrarily large distributional contrast. In practice, overly large distributional contrast may also introduce significant off-policy issues during training.

In Proposition 2, we assume $\bar{\pi}_t = \pi_t$ for all $t \geq 0$. An additional result with more moderate assumptions is the following.

Proposition 4. *Assume $\max_{t,i}\left|\frac{\bar{\pi}_t(i)}{\pi_t(i)} - 1\right| \leq \kappa$, and the vector steering is γ-good as defined in Proposition 2. Then, under the VSPO population score A_t^{VSPO} in Appendix E.3, we have $J(\pi_T) \geq r(i^\star) - \varepsilon$ when*

$$T \geq \frac{\lambda\Delta_{\max}}{2\eta\sqrt{1-\frac{1}{G}}\left(\left(1-\frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1-\frac{1}{G}\right)\kappa D_x\right)} \log\frac{C_0}{\varepsilon}, \tag{10}$$

where λ, D_x are defined in Eqn. (3) and assuming $\left(1-\frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1-\frac{1}{G}\right)\kappa D_x > 0$.

4 Experiments

4.1 Experimental Setup

4.1.1 VSPO Implementation Details

Steering Vector Construction. Since the base model may not reliably produce well-separated behavioral variants on its own, we construct the steering direction with the teacher-assisted pipeline summarized in Stage 1 of Figure 2. In our experiments, we use 200 positive-negative pairs per target behavior. After obtaining the layer-wise vector following [5], we select the strongest layer and normalize the resulting vector, which makes the steering coefficient comparable across tasks and behaviors. The complete construction procedure, is provided in Appendix G.1.1.

Self-Distillation Policy Optimization via Vector Steering. For rollout generation, we use $\mathcal{B} = \{-0.3, -0.15, 0, 0.15, 0.3\}$ and $\alpha = 0.1$ as the default across tasks, datasets, and models. Distinct symmetric coefficients provide local comparisons around the unsteered policy, while the combined task and behavioral reward determines each rollout's advantage. Sensitivity to steering magnitude, reward weight, and vector quality is reported in Appendix G.2.

4.1.2 Tasks and Settings

We evaluate our method on two complementary benchmarks: MMLU-Pro [25] and MATH [9]. MMLU-Pro covers a mixture of knowledge-intensive and reasoning tasks in a broad range, while MATH focuses on structured mathematical reasoning. This combination allows us to study controllability across both general-domain QA and step-by-step problem solving. Unless otherwise specified, all experiments are conducted with Qwen3-4B.

We consider three orthogonal but practically entangled control dimensions: preference-aligned reasoning behavior, robustness to misleading context, and concise reasoning. Each dimension captures a distinct aspect of output quality, yet none can be reliably reduced to another (e.g., conciseness alone cannot explain writing type and factual reliability).

- **Preference-Aligned Reasoning Behavior: Expertise Level.** We study controllability over the level of explanation, focusing on two levels. *Elementary-level* solutions are designed for readers with limited knowledge: they use simple language, short sentences, and explicit step-by-step reasoning,

carefully explaining each operation while avoiding advanced notations. In contrast, *expert-level* solutions target sophisticated readers: they use professional language, formal notation, and compact derivations, omitting routine intermediate steps and avoiding explanations of basic concepts.

- **Preference-Aligned Reasoning Behavior: Confidence Expression.** We control the stylistic presentation of confidence according to user preference. The *confident response* is written in a direct and decisive tone, using assertive language and minimal hedging. The *cautious response* is written in a calibrated tone, using uncertainty-aware phrasing such as "the best-supported answer is" or "based on the information given", while still committing to a final option.
- **Robustness to Misleading Context.** We evaluate whether models can maintain correct reasoning under explicitly misleading user-provided context. We use a diverse set of strong prompts to induce agreement pressure. Models are evaluated on their ability to produce the ground-truth answer despite this bias (independence) versus following the injected answer (agreement). This setting isolates the tension between correctness and context alignment, providing a controlled testbed for studying sycophancy and its mitigation.
- **Concise Reasoning:** Modern LLMs, particularly distilled small models, face the verbosity issue. Excessively long responses increase latency and cost, while overly short responses may omit critical reasoning steps. Our goal is to compress response length without sacrificing correctness. A central challenge is that the optimal length is instance-dependent: different questions require different levels of detail, and there is no global target length that is uniformly optimal.

We evaluate Preference-Aligned Reasoning Behavior using an LLM-based evaluator (Claude Sonnet 4.6 [2]), with prompts provided in Listing 3. More details are given in Appendix G.1.2. Only preference-aligned behavior scores use an LLM judge: misleading-context robustness uses accuracy and agreement rate, and concise reasoning uses accuracy and response length. Expertise comparisons across four judges are reported in Appendix I.

4.1.3 Baselines

We compare VSPO with representative baselines. Since some methods, such as SDFT and SDPO, were not designed for controllable model behavior, we adapt their data selection, prompting, or reward signals to include the target behavioral attribute. Additional details are provided in Appendix F. GRPO-family comparisons share the base model, batch size, rollout count, context window, and training steps; method-specific tuning and a prompt-policy mixture comparison are described in Appendices G and G.3.

- **Teacher-Trace Distillation (SFT).** Teacher-Trace Distillation (SFT) uses an external teacher model to generate reasoning traces aligned with target behaviors, such as concise reasoning or expert-level explanations. The student model is then trained on these traces using SFT. This follows the standard offline teacher-distillation paradigm [16] and is closely related to short-CoT distillation methods [8, 13].
- **Teacher-Trace Distillation (SFT) + GRPO.** We further evaluate a two-stage variant that first performs teacher-trace SFT and then applies GRPO. This follows the standard post-training recipe in which SFT provides an initial policy aligned with the desired response format or behavior, and RL further optimizes the model for task reward [6, 32].
- **Self-Distillation Fine-Tuning (SDFT).** SDFT [21] serves as a self-distillation baseline. SDFT uses the model itself as the teacher by generating training data under demonstration-conditioned prompting. In the experiments, we condition the demonstrations and prompts on the target behaviors.
- **Self-Distillation Policy Optimization (SDPO).** SDPO [10] improves a policy using high-quality rollouts sampled from the model itself. We adapt SDPO by selecting preferred rollouts according to both task correctness and the target behavioral attribute, and then optimize the model with a GRPO-style objective. This gives SDPO access to the same behavioral criterion used in our evaluation.
- **Reward-Shaped GRPO.** Reward shaping directly incorporates the target behavior into the GRPO reward. For concise reasoning, we add a length penalty to encourage shorter responses [23, 1, 32]. For preference-aligned reasoning behaviors, we use an LLM-based evaluator to reward responses that match the desired preference. For robustness to misleading context, we do not add an extra behavior-specific reward, since the target behavior is already captured by task accuracy.
- **GRPO with Off-Policy Textual Guidance.** We include a text-guided GRPO baseline inspired by methods that use auxiliary textual supervision, such as instructions or feedback, during RL training [30, 27, 22, 29]. During rollout generation, we append the target-behavior instruction to the orig-

Method	MMLU-Pro				MATH			
	Expert		Elementary		Expert		Elementary	
	Level ↑	Acc	Level ↓	Acc	Level ↑	Acc	Level ↓	Acc
Original model	26.3	44.0	26.3	44.0	26.5	75.1	26.5	75.1
Teacher-Trace Distillation (SFT)	45.0	66.3	21.4	64.9	43.1	74.0	22.4	73.9
Teacher-Trace Distillation (SFT) + GRPO	29.1	71.6	28.0	67.5	38.1	75.2	24.8	77.0
SDFT	33.4	61.3	20.7	59.3	40.6	76.3	23.1	76.9
SDPO	31.2	45.2	22.5	54.8	29.2	72.9	25.4	73.8
Reward-Shaped GRPO	30.6	63.5	25.8	60.1	27.0	75.2	24.3	73.7
GRPO with Off-Policy Textual Guidance.	30.5	59.2	26.1	52.9	39.8	75.3	20.1	76.4
VSPO (Ours)	**64.3**	**75.4**	**12.4**	**73.7**	**46.1**	**83.6**	**19.9**	**84.5**

Table 1: Preference-Aligned Reasoning Behavior: Expertise Level. We evaluate whether each method can adapt the explanation style to different explanation expertise preferences, on MMLU-Pro and MATH. "Level" measures explanation expertise, where higher values indicate more expert-like reasoning and lower values indicate more elementary reasoning. "Acc" denotes answer accuracy. VSPO achieves the strongest bidirectional expertise control, while maintaining high task accuracy on both datasets.

inal prompt. During optimization, we remove this instruction and compute policy log probabilities under the original prompt, updating the unprompted policy with the standard GRPO objective [19].

4.2 Main Results

Results on Preference-Aligned Reasoning Behavior. We evaluate two preference-aligned reasoning characteristics: expertise level and confidence expression. As shown in Table 1, existing baselines exhibit a clear trade-off between behavioral control and task accuracy. Some methods can shift the response style but degrade accuracy, such as Teacher-Trace Distillation and SDFT, while others preserve accuracy but fail to induce the desired behavior reliably, such as Teacher-Trace Distillation (SFT) + GRPO and Reward-Shaped GRPO. In contrast, VSPO achieves the strongest overall performance, steering responses toward the target expertise direction while preserving, and in some cases improving, task accuracy. We observe a similar trend for confidence expression in Figure 3. On Qwen3-8B, VSPO also improves MMLU-Pro expertise/accuracy from 30.3/72.8 to 45.1/74.6, providing initial scaling evidence beyond concise reasoning (Table 5).

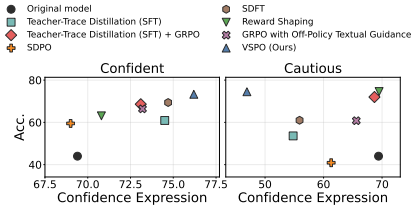


Figure 3: Results on MMLU-Pro for confident and cautious target behaviors. Higher confidence-expression scores indicate a more confident style; thus, upper-right is preferred for confident control, while upper-left is preferred for cautious control.

Across four LLM judges on MATH, VSPO receives higher expert scores and lower elementary scores than teacher-trace SFT. This consistent ordering supports the behavioral-control conclusion across evaluator families; complete scores are provided in Table 6.

Results on Robustness to Misleading Context. Figure 4 evaluates robustness to misleading context, where the desired behavior, reasoning independently rather than sycophancy, is directly measured by task accuracy. This setting should favor vanilla GRPO, since no separate behavior reward is needed: rollouts that resist the misleading context receive higher task reward. In practice, however, the original model is strongly biased by the misleading statement, achieving only 23.6% accuracy. As a result, many rollout groups contain no correct responses, leaving GRPO with sparse signal for escaping sycophancy. VSPO addresses this bottleneck by using vector steering to elicit more independent reasoning traces during rollout generation. These steered rollouts increase the probability that each group contains a correct response, yielding more informative relative advantages. As shown in Figure 4, VSPO achieves the highest final accuracy 64.2% and lower agreement rate. Its training curve starts from a higher critic score and improves faster, suggesting that latent steering exposes useful high-reward trajectories early and enables more stable optimization under misleading-context perturbations.

Results on Concise Reasoning. We evaluate concise reasoning with the goal of reducing verbosity without sacrificing accuracy. Following prior work, we use DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B, and compare against length-penalty reward shaping as the main baseline. Figure 5 reports the accuracy–length trade-off. Using the verbosity vectors from [3], VSPO consistently improves over both vanilla GRPO and reward shaping. Notably, on MMLU-Pro, these

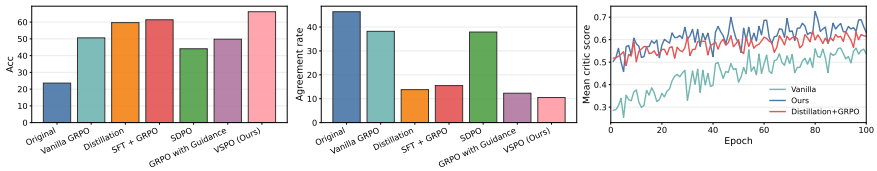


Figure 4: Results on robustness to misleading context. Left: task accuracy. Middle: agreement rate with the misleading context. Right: training dynamics measured by mean critic score. VSPO uses vector-steered rollouts to expose more independent reasoning traces, leading to higher final accuracy, lower agreement with the misleading context, and faster improvement in critic score.

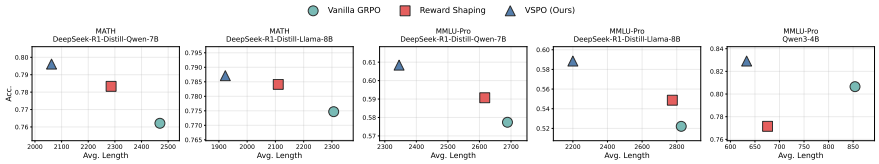


Figure 5: Accuracy-length trade-off for concise reasoning target behavior, on MATH and MMLU-Pro for different base models. VSPO consistently moves the model toward the upper-left region, achieving higher accuracy with shorter responses.

vectors remain effective despite being constructed from mathematical questions: VSPO achieves 60.8% accuracy with 2345.26 tokens, compared to 59.1% accuracy and 2616.70 tokens for reward shaping. With a Qwen3-4B verbosity vector constructed directly on MMLU-Pro, VSPO again obtains the best trade-off. Unlike length penalties, which only adjust rewards after generation, VSPO changes the rollout distribution by steering responses toward different verbosity levels. This structured exploration increases the chance of sampling concise yet correct traces, allowing GRPO to assign more informative advantages and distill shorter high-reward responses into the unsteered policy.

4.3 Understanding VSPO: Policy-Proximal Exploration, Curriculum Effects and Efficiency

Vector-steered rollouts remain policy-proximal. We evaluate each trace source using mean token-level NLL under the original unsteered model. Figure 6 shows lower NLL for vector-steered Qwen traces than prompting and rewriting baselines in both expertise settings, especially for expert-style explanations. The unsteered policy thus more readily models these steered traces. The analysis uses the largest steering coefficient in the training range.

Vector steering remains effective throughout training and induces an automatic curriculum. Figure 7(left) applies the same fixed expert-style steering vector to checkpoints from different stages of VSPO training. Despite changes in model weights, the vector continues to move generations in the target direction, with expertise-level scores increasing steadily over training. This indicates that the learned direction remains aligned with the evolving policy. This persistence induces an implicit curriculum. Early in training, steering exposes moderate, policy-close behavioral shifts; later, the same vector elicits stronger and more target-aligned traces. Consequently, VSPO obtains progressively more informative self-distillation signals without recomputing the vector or relying on additional external guidance, which helps stabilize optimization and improve sample efficiency.

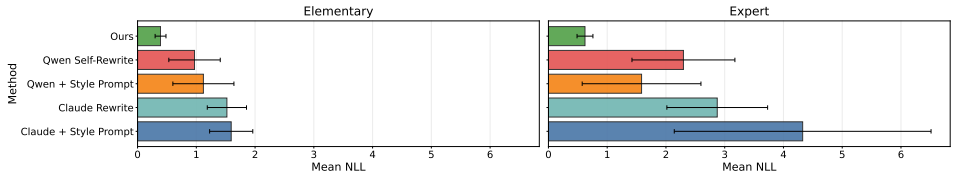


Figure 6: Reference-policy NLL by trace source. We report mean token-level negative log-likelihood under the original unsteered model. Vector-steered traces have lower NLL than prompted or rewritten traces in both settings.

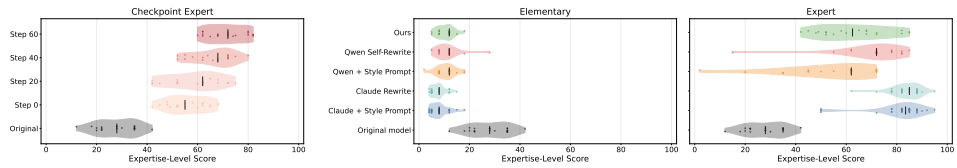


Figure 7: Training dynamics and behavior-space coverage induced by vector steering. Left: expertise-level score distributions obtained by applying the same fixed expert-style steering vector to model checkpoints at different VSPO training stages. The shift toward higher scores shows that the vector remains effective with training. Middle and right: expertise-level score distributions for different trace sources in the elementary and expert settings. VSPO produces structured shifts toward the target direction. By contrast, other baselines often provide unreliable control, either collapsing to narrow regions or producing noisy/misaligned samples.

Vector steering enables structured behavior exploration for more informative policy updates. Figure 7(middle and right) compares expertise-score distributions across trace sources. Ordinary sampling occupies a narrow style region, while prompting and rewriting can produce narrow or misaligned shifts. Vector steering exposes responses across distinct intensities in the intended direction, providing task- and behavior-dependent candidates for GRPO. A direct prompt-policy mixture with positive, negative, and no behavioral instructions reaches 28.4 expertise and 75.9% accuracy on MATH, compared with VSPO's 46.1 and 83.6% under the same rollout and update budget (Table 5). VSPO's improvements in both metrics support targeted latent exploration.

Sensitivity to vector quality and hyperparameters. Table 2 evaluates Qwen3-4B on MATH, using $\mathcal{B}(b) = \{-b, -b/2, 0, b/2, b\}$. With 50 rather than 200 contrastive pairs, expertise control weakens while accuracy remains above the original model. Halving b or doubling α yields similar outcomes within the tested range; further details are in Appendix G.2.

Setting	Pairs	b	α	Expertise ↑	Accuracy (%) ↑
Original model	–	–	–	26.5	75.1
VSPO (default)	200	0.30	0.1	46.1	83.6
Fewer contrastive pairs	50	0.30	0.1	39.3	82.7
Smaller steering range	200	0.15	0.1	46.0	83.8
Larger reward weight	200	0.30	0.2	45.8	83.2

Table 2: Sensitivity of MATH expertise control to contrastive-pair count, steering range, and behavioral reward weight. Each VSPO variant changes one setting relative to the default; all use the symmetric coefficient set $\mathcal{B}(b)$.

We provide additional **efficiency** discussion in Appendix H.

5 Discussion

We introduced VSPO as a method for learning controllable model behaviors while preserving model accuracy. By using steering vectors to generate on-steered-policy rollouts with varying intensities, VSPO alleviates sparse behavioral rewards and provides more informative candidates for GRPO-style optimization. Across explanation expertise, confidence expression, robustness to misleading context, and verbosity control, VSPO improves target control while maintaining or improving accuracy compared with reward shaping, teacher-guided distillation, and self-distillation baselines. A key limitation is that VSPO depends on the quality and alignment of the learned steering vector: if the vector is weak or fails to expose useful high-reward variants, the theoretical and empirical benefits may diminish. Our experiments optimize task correctness together with one behavioral attribute at a time; simultaneous control of conflicting attributes remains future work. Behavioral scores also depend on LLM evaluators, and strong steering misalignment can disrupt generation. Future work includes more robust and fully self-guided vector construction, together with joint optimization of multiple behavioral attributes across a broader range of tasks.

Acknowledgments

This work is supported in part by the NSF grants CCF-2550179, CCF-2403075, and CCF-2212426, and by the ONR grant N00014-24-1-2289. Computational aspects of the research is generously supported by an Amazon Research Award on Foundation Model Development.

References

[1] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Second Conference on Language Modeling*.

[2] Anthropic. Claude sonnet 4.6 system card. https://anthropic.com/claude-sonnet-4-6-system-card, 2026.

[3] Seyedarmin Azizi, Erfan Baghaei Potraghloo, Souvik Kundu, and Massoud Pedram. Activation steering for chain-of-thought compression. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

[4] Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. Understanding (un) reliability of steering vectors in language models. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

[5] Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

[6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[7] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

[8] Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025.

[9] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[10] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

[11] Muhammed Emrullah Ildiz, Halil Alperen Gozeten, Ege Onur Taga, and Samet Oymak. Learning to correct: Calibrated reinforcement learning for multi-attempt chain-of-thought. *International Conference on Machine Learning*, 2026.

[12] Shawn Im and Sharon Li. A unified understanding and evaluation of steering methods. *arXiv preprint arXiv:2502.02716*, 2025.

[13] Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320, 2025.

[14] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. In *Forty-second International Conference on Machine Learning*.

[15] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning*, pages 26874–26901. PMLR, 2024.

[16] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, 2025.

[17] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[18] Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

[19] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[20] Archit Sharma, Sedrick Keh, Eric Mitchell, Chelsea Finn, Kushal Arora, and Thomas Kollar. A critical evaluation of ai feedback for aligning large language models. *Advances in Neural Information Processing Systems*, 37:29166–29190, 2024.

[21] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

[22] Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

[23] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[24] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[25] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[26] Nuoya Xiong and Aarti Singh. Projection optimization: A general framework for multi-objective and multi-group rlhf. In *Forty-second International Conference on Machine Learning*.

[27] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

[28] Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *International Conference on Machine Learning*, pages 56276–56297. PMLR, 2024.

[29] Zhe Yang, Yudong Wang, Rang Li, and Zhifang Sui. Towards better rl training data utilization via second-order rollout. *arXiv preprint arXiv:2602.22765*, 2026.

[30] Jiashu Yao, Heyan Huang, Shuang Zeng, Chuwei Luo, Wangjie You, Jie Tang, Qingsong Liu, Yuhang Guo, and Yangyang Kang. Incorporating self-rewriting into large language model reasoning reinforcement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34405–34413, 2026.

[31] Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning. *Advances in Neural Information Processing Systems*, 38:96726–96752, 2026.

[32] Xuechen Zhang, Zijian Huang, Chenshun Ni, Ziyang Xiong, Jiasi Chen, and Samet Oymak. Making small language models efficient reasoners: Intervention, supervision, reinforcement. *arXiv preprint arXiv:2505.07961*, 2025.

[33] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

[34] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Appendix

The appendix is organized as follows:

- Appendix A describes our usage of LLMs.
- Appendix B provides the pseudocode for VSPO.
- Appendix C gives additional related work.
- Appendix D presents further methodological details.
- Appendix E contains the full theoretical analysis and proofs.
- Appendix F describes the baseline implementations.
- Appendix G provides experimental details, including compute resources and evaluation protocols.
- Appendix H discusses the efficiency of VSPO.
- Appendix I studies robustness to the choice of LLM evaluator.
- Appendix J and Appendix K provide the prompts and example responses used in our experiments.

A Usage of LLMs

In our experiments, we use LLMs as evaluators for preference-aligned reasoning behavior and as trace generators or rewriters in several baselines. Specifically, we use Claude Sonnet 4.6 and Qwen3-4B for style evaluation, with additional checks using GPT-5.4 and Gemini-3.5-Flash for expertise scoring, and Claude Opus 5 for a blinded pairwise reliability check, and use Claude Sonnet 4.6 and Qwen3-4B for teacher/rewrite baselines. Task accuracy is computed using task-specific answer extraction and matching rather than LLM judgments. LLMs were also used for writing assistance.

B Pseudo Code

Algorithm 1 VSPO: Vector-Steered Policy Optimization

Require: Initial policy π_θ, reference policy π_{ref}, stronger teacher model $\mathcal{T}$, steering intensity set $\mathcal{B} = \{\beta_1, \ldots, \beta_G\}$, reward weight α

Stage 1: Vector construction
Sample a set of prompts $\{q_i\}_{i=1}^N$ for vector construction
for each prompt q_i **do**
  Generate a base response $o_i \sim \pi_\theta(\cdot \mid q_i)$
  Use teacher $\mathcal{T}$ to rewrite o_i into two opposite directions of the target property:
    positive rewrite $o_i^+ \leftarrow \mathcal{T}(q_i, o_i, \text{positive direction})$
    negative rewrite $o_i^- \leftarrow \mathcal{T}(q_i, o_i, \text{negative direction})$
end for
Form contrastive sets $\mathcal{D}^+ = \{(q_i, o_i^+)\}_{i=1}^N$ and $\mathcal{D}^- = \{(q_i, o_i^-)\}_{i=1}^N$
Extract activations and compute steering vector v^ℓ using Eqn. (12)
Optionally normalize v^ℓ and select the best layer ℓ on validation data
Stage 2: GRPO with vector steering
for each training iteration **do**
  Sample a batch of prompts $\{q_i\}_{i=1}^B$
  for each prompt q_i **do**
    for each $\beta_g \in \mathcal{B}$ **do**
      Generate steered rollout $o_{i,k} \sim \tilde{\pi}_{\theta_{\text{old}}}^{(\beta_g)}(\cdot \mid q_i)$
        by injecting $\tilde{h}_t^\ell \leftarrow h_t^\ell + \beta_g v^\ell$
      Compute reward

$$r_{i,g} = r_0(q_i, o_{i,g}) + \alpha\beta_g$$

    end for
    Compute group-relative advantages $\{A_{i,g}\}_{g=1}^G$ from $\{r_{i,g}\}_{g=1}^G$
  end for
  Recompute token probabilities using the unsteered current policy π_θ
  Update θ by maximizing Eqn. (1)
end for

C Related Work

Teacher-Trace Distillation and SFT. A common alternative to RL is to use a stronger teacher model to generate reasoning traces and then train the target model with supervised fine-tuning. This paradigm has been widely used for improving reasoning performance and controlling response format, including recent work on distilling short or compressed chain-of-thought traces to reduce inference length [8, 13, 16]. In our setting, such methods can be adapted by asking the teacher to produce traces aligned with a desired behavior, such as expert-level explanations or concise reasoning. However, teacher-trace SFT is fundamentally off-policy: the student learns from trajectories generated by an external model rather than from its own sampled rollouts. As a result, the distilled behavior may not match the student policy's reachable trajectory distribution, and further RL is often needed to recover task performance or adapt the learned behavior to the student model [6, 32].

RL Post-Training for Reasoning. Reinforcement learning has become a central tool for post-training LLMs, especially for reasoning tasks where rewards can be derived from verifiable correctness. Early RLHF methods optimize policies against learned preference rewards, while recent RLVR methods use task-specific verifiers to provide more direct supervision for mathematical and code reasoning. GRPO is particularly relevant to our setting because it removes the need for a learned critic and computes relative advantages within a group of sampled responses, making it a practical optimization method for reasoning models [19, 6]. However, standard GRPO primarily improves task performance through scalar reward signals and ordinary rollout sampling. When the desired behavior involves both correctness and a secondary characteristic, such as conciseness, confidence expression, or resistance to sycophantic bias, vanilla rollout sampling may not expose sufficiently diverse or high-quality behavioral variants for the group-relative objective to exploit.

Self-Distillation for LLM Post-Training. Recent self-distillation methods aim to reduce the off-policy mismatch of teacher-trace training by deriving learning signals from the model itself. SDFT uses the model in a demonstration-conditioned mode as a self-teacher, producing distillation targets from the model's own conditional distribution rather than from fixed external traces [21]. SDPO further extends this idea to policy optimization by using high-quality self-generated rollouts as implicit training targets and optimizing them with a GRPO-style objective [10]. These methods are closely related to our motivation, since they treat the current model as a source of supervision rather than relying entirely on an external teacher. However, they are primarily designed for improving task performance, not for learning controllable secondary characteristics. In contrast, our method uses vector steering to systematically reveal behaviorally diverse variants of the current policy along a target latent direction, and then distills the useful variants back into the unsteered model through RL.

Reward Shaping and Multi-Objective RL. Another natural approach is to incorporate the target characteristic directly into the reward. For example, concise reasoning can be encouraged with a length penalty, while preference-aligned reasoning behavior can be rewarded with an LLM-based evaluator [23, 1, 32]. More broadly, multi-objective alignment methods study how to optimize LLMs under multiple preference dimensions or dynamically trade off competing objectives. These approaches are simple and often compatible with existing GRPO pipelines, but they only modify the scalar feedback after rollouts have already been generated. Therefore, if the rollout group lacks high-quality or behaviorally diverse candidates, reward shaping alone may provide a weak learning signal. Our method instead shapes the rollout distribution itself: vector steering actively elicits a structured set of behavioral variants before reward assignment, giving group-relative optimization more informative candidates to compare.

Textual Guidance and AI Feedback During RL. Several recent methods augment RL training with additional textual guidance, such as instructions, critiques, feedback, or AI-generated supervision, to elicit desired behaviors during rollout generation [22, 30, 27, 29]. Related RLAIF-style approaches use powerful models as judges or feedback providers, reducing the need for human annotation but often requiring repeated model calls during training. These methods can improve controllability, but their reliance on external textual guidance or online feedback can be expensive and may introduce a mismatch between guided rollout generation and unguided inference. Our method shares the goal of eliciting more useful rollouts during RL, but replaces repeated textual guidance with a reusable latent steering vector. This makes the intervention cheaper to apply and less dependent on surface-form prompting, while still allowing the final policy to be optimized and evaluated without additional guidance at inference time.

Persona Vectors and Vector Steering. Persona vectors and vector steering are most closely related to the literature on *representation engineering*, which studies how semantically meaningful directions in a model's internal activation space can be identified and manipulated to control behavior at inference time [34]. Early activation-steering methods, such as Activation Addition (ActAdd) and Contrastive Activation Addition (CAA), showed that steering vectors extracted from contrastive prompts or examples can reliably shift model outputs toward target properties without updating model weights, providing a lightweight alternative to fine-tuning for controllability and alignment [24, 18]. Building on this line of work, persona vectors specialize such latent directions to behavioral traits or character tendencies, such as sycophancy, hallucination, or harmfulness, and have been used both to monitor personality shifts and to intervene on them during or after training [5]. This perspective is appealing because it treats behavior as at least partially organized in representation space, making control more modular and interpretable than purely prompt-level or objective-level interventions [34, 5]. At the same time, recent work has emphasized that steering effects can be highly sensitive to vector construction, layer choice, and steering magnitude, and may be entangled across multiple attributes, which limits the robustness of simple inference-time activation editing [4, 12]. In contrast to prior work that primarily applies vector steering only at inference time, our method integrates vector steering into GRPO training, using steering-induced variation during policy optimization to shape the model's latent behavioral distribution and thereby produce more persistent and controllable adaptation.

D Method

We propose VSPO, a latent self-distillation algorithm that uses steered model generations for policy optimization. Our starting point is the observation that the current policy already contains diverse behavioral variants, but standard RL only observes them through ordinary sampling and scalar rewards. This creates a bottleneck: the policy may contain useful behaviors that are difficult to elicit reliably, and scalar reward alone does not determine how the policy should move toward them. Our key idea is to use vector steering to expose such behaviors in a controlled way, and then use RL to distill them back into the unsteered policy.

The key object in our method is the *steered policy*

$$\tilde{\pi}_\theta^{(\beta)}(\cdot \mid x), \tag{11}$$

which denotes the current policy under a latent perturbation of strength β along a learned steering direction. For a prompt x, the steered policy generates a rollout that can be viewed as a behaviorally modified variant of what the current model would produce on its own. By evaluating multiple such variants within each training group, we obtain a structured set of on-steered-policy candidates that differ along a targeted behavior axis. The role of RL is then to reinforce the variants that best improve the task objective and thereby distill this behavior back into the base policy.

Algorithm 1 summarizes the full pipeline. First, we construct a steering vector that captures a target behavior in latent space. Second, for each prompt, we sample a group of rollouts under different steering intensities. Third, we score these rollouts with a reward that combines the primary task signal with a steering-dependent preference term, and update the unsteered policy using a GRPO-style objective. In this way, VSPO turns vector steering into a mechanism for self-distillation: steering reveals candidate behaviors from the current model, and policy optimization internalizes the useful ones.

D.1 Vector Construction

We first construct a steering vector v^ℓ at a transformer layer ℓ that captures a target behavioral direction. At a high level, we obtain this direction from contrastive examples that differ mainly in the property of interest. In our setting, these contrastive examples are produced with teacher assistance: for prompts sampled from the training distribution, we first generate responses from the current small model, and then use a stronger model(Claude Sonnet 4.6) to rewrite them toward two opposite directions of the target property. We then compute a mean activation difference between the resulting positive and negative examples to obtain

$$v^\ell = \frac{1}{N}\sum_{i=1}^{N} \phi^\ell(s_i^+) - \frac{1}{N}\sum_{i=1}^{N} \phi^\ell(s_i^-), \tag{12}$$

where $\phi^{\ell}(\cdot)$ denotes an activation summary at layer ℓ, N is the number of pairs, s_i^+ is the positive sample and s_i^- is the corresponding negative sample. We optionally normalize the vector and select the best layer on held-out data. The details of vector construction are deferred to Section 4.1.1.

The purpose of this stage is not to produce supervised targets for policy learning. Rather, it identifies a latent direction along which the current policy can be systematically perturbed during training. This keeps the teacher's role limited: the teacher helps define the behavior axis, while the actual learning still happens through the model's own rollouts.

D.2 Self-Distillation Policy Optimization via Vector Steering

We now describe how vector steering induces self-distillation during RL. For each prompt x, instead of sampling an unstructured group of rollouts, we generate one rollout for each steering intensity in a fixed set

$$\mathcal{B} = \{\beta_1, \ldots, \beta_G\}. \tag{13}$$

For example, $\mathcal{B}$ may be chosen as $\{-a, -b, 0, b, a\}$. Given a steering intensity β_g, we inject the steering vector into the residual stream during decoding,

$$\tilde{h}_t^{\ell} = h_t^{\ell} + \beta_g v^{\ell}, \tag{14}$$

and sample

$$o_k \sim \tilde{\pi}_{\theta_{\text{old}}}^{(\beta_g)}(\cdot \mid q). \tag{15}$$

The model samples these on-steered-policy rollouts from $\tilde{\pi}_{\theta_{\text{old}}}^{(\beta_g)}$ and computes policy updates with the unsteered model. The steered policy serves as a self-teacher by exposing behavioral variants of the current model. The external teacher defines the behavior axis through contrastive rewrites, while RL distills the model's own steered generations into the base policy.

Each rollout is assigned reward

$$r_g = r_0(q, o_g) + \alpha\beta_g, \tag{16}$$

where r_0 is the primary task reward and α controls the steering preference. Let $\bar{r}$ and σ_r be the mean and standard deviation of the group of rewards $\{r_k\}_{g=1}^{G}$, where G is the number of samples in the same GRPO group. The advantge of each rollout is defined as

$$A_g = \frac{r_g - \bar{r}}{\sigma_r}. \tag{17}$$

A key design choice is that we use steering only for sampling, not for the policy update itself. For a rollout $y_g = (y_{g,1}, \ldots, y_{g,T_g})$, where T_g is the token number of the rollout y_g, we compute the standard token-level ratio under the unsteered policy,

$$\rho_{g,t}(\theta) = \frac{\pi_\theta(o_{g,t} \mid q, o_{g,<t})}{\pi_{\theta_{\text{old}}}(o_{g,t} \mid q, o_{g,<t})}, \tag{18}$$

with no steering in either numerator or denominator. The resulting objective is

$$\mathcal{L}_{VSPO}(\theta) = \mathbb{E}_x \left[\frac{1}{G} \sum_{g=1}^{G} \frac{1}{T_g} \sum_{t=1}^{T_g} \min\big(\rho_{g,t}(\theta)A_g,\ \text{clip}\big(\rho_{g,t}(\theta), 1-\epsilon, 1+\epsilon\big)A_g\big) \right] - \lambda \mathcal{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}). \tag{19}$$

where clip is the clip function, ϵ is the clip ratio, $\mathcal{D}_{\text{KL}}$ is the KL divergence, and π_{ref} is the reference policy. This objective has a simple interpretation. Vector steering changes the rollout distribution so that the current policy reveals diverse but targeted behavioral variants. GRPO then selects which of these variants should be preferred and updates the unsteered policy accordingly. Thus, VSPO performs self-distillation in latent space: the model learns from steered variants of its own behavior, rather than from fully off-policy teacher traces.

Discussion. Our method is closely aligned with the spirit of self-distillation policy optimization. The main difference is the source of the distillation signal. In SDPO, the self-teacher arises by conditioning the current model on rich feedback and re-evaluating the same rollout, yielding a denser signal for credit assignment. In our setting, the self-teacher arises by steering the current model along a learned latent direction, yielding a controlled family of on-steered-policy behavioral variants. In both cases, the current policy provides the source of supervision, and policy optimization distills that information back into the model. The advantage of our formulation is that it does not require tokenized environment feedback; instead, it uses latent steering to generate diverse and controllable rollouts directly during sampling, while remaining compatible with standard GRPO-style training.

E More on Theoretical Analysis

E.1 More on Theoretical Setup

Lemma 5. *Define the scalarized reward $r(i) = x(i) + \alpha y(i)$ where $\alpha > 0$. If*

$$\alpha > \max_{i:\, y(i)<y(i^\star)} \frac{\max\{x(i) - x(i^\star), 0\}}{y(i^\star) - y(i)},$$

then $i^\star$ defined as

$$i^\star = \arg\max_{i\in[K]} \left\{ x(i) : y(i) = \max_{j\in[K]} y(j) \right\}$$

is the unique maximizer of the scalarized reward $r(i)$, i.e., $i^\star = \arg\max_{i\in[K]} r(i)$.

Proof. For any arm $i \neq i^\star$, if $y(i) = y(i^\star)$, then we have $r(i^\star) > r(i)$ since $x(i^\star) > x(i)$.

If $y(i) < y(i^\star)$ and $x(i) \leq x(i^\star)$, then we also have $r(i^\star) = x(i^\star) + \alpha y(i^\star) > x(i) + \alpha y(i) = r(i)$.

If $y(i) < y(i^\star)$ and $x(i) > x(i^\star)$, then by the condition of α we have

$$\alpha > \frac{x(i) - x(i^\star)}{y(i^\star) - y(i)} \implies \alpha y(i^\star) - \alpha y(i) > x(i) - x(i^\star),$$

which implies $r(i^\star) = x(i^\star) + \alpha y(i^\star) > x(i) + \alpha y(i) = r(i)$, as desired. □

E.2 Population Advantage Score for GRPO

Lemma 6 (GRPO Populations)**.** *Define*

$$\mu_{r,t} := \sum_{j=1}^{K} \pi_t(j) r(j),\ \sigma^2_{r,t} := \sum_{j=1}^{K} \pi_t(j) \big(r(j) - \mu_{r,t}\big)^2.$$

Then, for every arm i,

$$\mathbb{E}\left[N_t(i)(r(i) - \bar{r}_t^{\mathrm{GRPO}}) \,\middle|\, \pi_t\right] = (G-1)\pi_t(i)\big(r(i) - \mu_{r,t}\big),\ \mathbb{E}\left[\big(\bar{\sigma}_t^{\mathrm{GRPO}}\big)^2 \,\middle|\, \pi_t\right] = \sigma^2_{r,t}.$$

Proof. Since the G samples are i.i.d. from π_t, we have

$$\mathbb{E}[N_t(i) \mid \pi_t] = G\pi_t(i).$$

It remains to compute $\mathbb{E}[N_t(i)\bar{r}_t^{\mathrm{GRPO}} \mid \pi_t]$. Since

$$N_t(i) = \sum_{g=1}^{G} \mathbf{1}\left\{a_t^{(g)} = i\right\},\ \bar{r}_t^{\mathrm{GRPO}} = \frac{1}{G}\sum_{h=1}^{G} r\left(a_t^{(h)}\right),$$

we have

$$\mathbb{E}[N_t(i)\bar{r}_t^{\mathrm{GRPO}} \mid \pi_t] = \frac{1}{G}\sum_{g=1}^{G}\sum_{h=1}^{G} \mathbb{E}\left[\mathbf{1}\left\{a_t^{(g)} = i\right\} r\left(a_t^{(h)}\right) \,\middle|\, \pi_t\right].$$

Since

$$\mathbb{E}\left[\mathbf{1}\left\{a_t^{(g)}=i\right\} r\left(a_t^{(h)}\right) \,\middle|\, \pi_t\right] = \begin{cases} \pi_t(i)r(i), & g=h, \\ \pi_t(i)\mu_{r,t}, & g\neq h, \end{cases}$$

we have

$$\mathbb{E}[N_t(i)\bar{r}_t^{\text{GRPO}} \mid \pi_t] = \frac{1}{G}\left(G\pi_t(i)r(i) + G(G-1)\pi_t(i)\mu_{r,t}\right) = \pi_t(i)r(i) + (G-1)\pi_t(i)\mu_{r,t}.$$

This implies

$$\mathbb{E}\left[N_t(i)(r(i)-\bar{r}_t^{\text{GRPO}}) \,\middle|\, \pi_t\right] = \mathbb{E}[N_t(i) \mid \pi_t]r_i - \mathbb{E}[N_t(i)\bar{r}_t^{\text{GRPO}} \mid \pi_t] = (G-1)\pi_t(i)(r(i)-\mu_{r,t}),$$

as desired.

On the other hand, notice that

$$\begin{aligned}\left(\bar{\sigma}_t^{\text{GRPO}}\right)^2 &= \frac{1}{G-1}\sum_{g=1}^{G}\left(r\left(a_t^{(g)}\right) - \bar{r}_t^{\text{GRPO}}\right)^2 \\ &= \frac{1}{G-1}\left(\sum_{g=1}^{G} r\left(a_t^{(g)}\right)^2 - G\left(\frac{1}{G}\sum_{g=1}^{G} r\left(a_t^{(g)}\right)\right)^2\right).\end{aligned}$$

By linearity of expectation and independce, We have

$$\mathbb{E}\left[r\left(a_t^{(g)}\right)^2 \,\middle|\, \pi_t\right] = \sigma_{r,t}^2 + \mu_{r,t}^2,$$

and

$$\mathbb{E}\left[\frac{1}{G}\sum_{g=1}^{G} r\left(a_t^{(g)}\right) \,\middle|\, \pi_t\right] = \mu_{r,t},\ \text{Var}\left[\frac{1}{G}\sum_{g=1}^{G} r\left(a_t^{(g)}\right) \,\middle|\, \pi_t\right] = \frac{\sigma_{r,t}^2}{G}.$$

This implies

$$\mathbb{E}[\left(\bar{\sigma}_t^{\text{GRPO}}\right)^2 \mid \pi_t] = \frac{G}{G-1}\left(\sigma_{r,t}^2 + \mu_{r,t}^2\right) - \frac{G}{G-1}\left(\mu_{r,t}^2 + \frac{\sigma_{r,t}^2}{G}\right) = \sigma_{r,t}^2,$$

as desired. □

Remark 7. In Proposition 1, we use the GRPO population arm-level score as

$$A_t^{\text{GRPO}}(i) := \frac{(G-1)\pi_t(i)(r(i)-\mu_{r,t})}{\sigma_{r,t}}. \tag{20}$$

E.3 Population Advantage Score for VSPO

Lemma 8 (VSPO Populations)**.** *Define* $\mu_{x,t}^{\pm} := \sum_{j=1}^{K}\pi_t^{\pm}(j)x(j)$; $\mu_{x,t} := \frac{\mu_{x,t}^{+}+\mu_{x,t}^{-}}{2}, \mu_{y,t} := \frac{\mu_{y,t}^{+}+\mu_{y,t}^{-}}{2}; \mu_t^{\pm} := \mu_{x,t}^{\pm} + \alpha\mu_{y,t}^{\pm}; \mu_t = \frac{\mu_t^{+}+\mu_t^{-}}{2}; d_t(i) = \frac{\pi_t^{+}(i)-\pi_t^{-}(i)}{2}; \delta_{x,t} = \mu_{x,t}^{+} - \mu_{x,t}^{-}, \delta_{y,t} = \mu_{y,t}^{+} - \mu_{y,t}^{-}, \delta_t = \delta_{x,t} + \alpha\delta_{y,t} = \mu_t^{+} - \mu_t^{-}$, *and*

$$v_{r,t}^2 := \frac{1}{2}\sum_{i=1}^{K}\pi_t^{+}(i)\left(r_t^{+}(i)-\mu_t\right)^2 + \frac{1}{2}\sum_{i=1}^{K}\pi_t^{-}(i)\left(r_t^{-}(i)-\mu_t\right)^2,$$

Then, for every arm i,

$$\begin{aligned}&\mathbb{E}\Big[N_t^{+}(i)\big(r_t^{+}(i)-\bar{r}_t^{\text{VSPO}}\big) + N_t^{-}(i)\big(r_t^{-}(i)-\bar{r}_t^{\text{VSPO}}\big) \,\Big|\, \pi_t^{+}, \pi_t^{-}\Big] \\ &\qquad = (G-1)\bar{\pi}_t(i)\left(x(i)-\mu_{x,t}\right) + \frac{d_t(i)}{2}\left(\delta_t + \alpha(G-1)\delta_{y,t}\right), \\ &\mathbb{E}\Big[\left(\bar{\sigma}_t^{\text{VSPO}}\right)^2 \,\Big|\, \pi_t^{+}, \pi_t^{-}\Big] = v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}.\end{aligned}$$

Proof. Let $m := G/2$. At iteration t, applying vector steering returns π_t^+ and π_t^- such that $\bar{\pi}_t = \frac{\pi_t^+ + \pi_t^-}{2}$. Let

$$S_t^+ := \sum_{g=1}^{m} r_+\left(a_{t,+}^{(g)}\right),\ S_t^- := \sum_{g=1}^{m} r_-\left(a_{t,-}^{(g)}\right).$$

Then

$$N_t(i) = N_t^+(i) + N_t^-(i),\ \bar{r}_t^{\text{VSPO}} = \frac{S_t^+ + S_t^-}{G}.$$

By definitions we have

$$\pi_t^{\pm}(i) = \bar{\pi}_t(i) \pm d_t(i),\ \mu_t^{\pm} = \mu_t \pm \frac{\delta_t}{2}.$$

We first compute $\mathbb{E}\left[(N_t^+(i) + N_t^-(i))\ \bar{r}_t^{\text{VSPO}} \mid \pi_t^+, \pi_t^-\right]$. We have

$$\begin{aligned}
&\mathbb{E}\left[(N_t^+(i) + N_t^-(i))\ \bar{r}_t^{\text{VSPO}} \mid \pi_t^+, \pi_t^-\right] \\
&= \frac{1}{G}\mathbb{E}\left[(N_t^+(i) + N_t^-(i))(S_t^+ + S_t^-) \mid \pi_t^+, \pi_t^-\right] \\
&= \frac{1}{G}\left(\mathbb{E}[N_t^+(i)S_t^+] + \mathbb{E}[N_t^-(i)S_t^-] + \mathbb{E}[N_t^+(i)S_t^-] + \mathbb{E}[N_t^-(i)S_t^+]\right).
\end{aligned}$$

Similar to the proof of Lemma 6 yields

$$\begin{aligned}
\mathbb{E}[N_t^+(i)S_t^+] &= m\pi_t^+(i)r_t^+(i) + m(m-1)\pi_t^+(i)(\mu_{x,t}^+ + \alpha\mu_{y,t}^+) \\
&= m\pi_t^+(i)x(i) + m(m-1)\pi_t^+(i)\mu_{x,t}^+ + m^2\pi_t^+(i)\alpha\mu_{y,t}^+ \\
\mathbb{E}[N_t^-(i)S_t^-] &= m\pi_t^-(i)r_t^-(i) + m(m-1)\pi_t^-(i)(\mu_{x,t}^- + \alpha\mu_{y,t}^-) \\
&= m\pi_t^-(i)x(i) + m(m-1)\pi_t^-(i)\mu_{x,t}^- + m^2\pi_t^-(i)\alpha\mu_{y,t}^-
\end{aligned}$$

By independence we have

$$\mathbb{E}[N_t^+(i)S_t^-] = m^2\pi_t^+(i)(\mu_{x,t}^- + \alpha\mu_{y,t}^-),\ \mathbb{E}[N_t^-(i)S_t^+] = m^2\pi_t^-(i)(\mu_{x,t}^+ + \alpha\mu_{y,t}^+).$$

Combining these terms yields

$$\begin{aligned}
\mathbb{E}\left[(N_t^+(i) + N_t^-(i))\ \bar{r}_t^{\text{VSPO}} \mid \pi_t^+, \pi_t^-\right] = \frac{1}{G}\Big(&m(\pi_t^+(i) + \pi_t^-(i))x(i) \\
&+ m(m-1)\left(\pi_t^+(i)\mu_{x,t}^+ + \pi_t^-(i)\mu_{x,t}^-\right) \\
&+ m^2\left(\pi_t^+(i)\mu_{x,t}^- + \pi_t^-(i)\mu_{x,t}^+\right) \\
&+ m^2\alpha(\pi_t^+(i) + \pi_t^-(i))(\mu_{y,t}^+ + \mu_{y,t}^-)\Big).
\end{aligned}$$

Using the facts that

$$\begin{aligned}
&\pi_t^+(i) + \pi_t^-(i) = 2\bar{\pi}_t(i), \\
&\pi_t^+(i)\mu_{x,t}^+ + \pi_t^-(i)\mu_{x,t}^- = 2\bar{\pi}_t(i)\mu_{x,t} + d_t(i)\delta_{x,t}, \\
&\pi_t^+(i)\mu_{x,t}^- + \pi_t^-(i)\mu_{x,t}^+ = 2\bar{\pi}_t(i)\mu_{x,t} - d_t(i)\delta_{x,t}, \\
&\mu_{y,t}^+ + \mu_{y,t}^- = 2\mu_{y,t},
\end{aligned}$$

and $G = 2m$, we obtain

$$\mathbb{E}\left[(N_t^+(i) + N_t^-(i))\ \bar{r}_t^{\text{VSPO}} \mid \pi_t^+, \pi_t^-\right] = \pi_t(i)x(i) + (G-1)\bar{\pi}_t(i)\mu_{x,t} - \frac{1}{2}d_t(i)\delta_{x,t} + \alpha G\bar{\pi}_t(i)\mu_{y,t}.$$

Since

$$\mathbb{E}[N_t^+(i) \mid \pi_t^+, \pi_t^-] = m\pi_t^+(i),\ \mathbb{E}[N_t^-(i) \mid \pi_t^+, \pi_t^-] = m\pi_t^-(i),$$

we have

$$\begin{aligned}
\mathbb{E}[N_t^+(i)r_t^+(i) + N_t^-(i)r_t^-(i)] &= m\pi_t^+(i)[x(i) + \alpha\mu_{y,t}^+] + m\pi_t^-(i)[x(i) + \alpha\mu_{y,t}^-] \\
&= G\bar{\pi}_t(i)x(i) + G\alpha\left(\bar{\pi}_t(i)\mu_{y,t} + \frac{1}{2}d_t(i)\delta_{y,t}\right)
\end{aligned}$$

we conclude

$$\begin{aligned}
&\mathbb{E}\left[N_t^+(i)(r_t^+(i)-\bar{r}_t^{\text{VSPO}})+N_t^-(i)(r_t^-(i)-\bar{r}_t^{\text{VSPO}})\;\middle|\;\pi_t^+,\pi_t^-\right]\\
&=(G-1)\bar{\pi}_t(i)(x(i)-\mu_{x,t})+\frac{d_t(i)\delta_{x,t}}{2}+\frac{\alpha G d_t(i)\delta_{y,t}}{2}\\
&=(G-1)\bar{\pi}_t(i)(x(i)-\mu_{x,t})+\frac{d_t(i)}{2}\left(\delta_{x,t}+\alpha G\delta_{y,t}\right)\\
&=(G-1)\bar{\pi}_t(i)(x(i)-\mu_{x,t})+\frac{d_t(i)}{2}\left[\delta_t+\alpha(G-1)\delta_{y,t}\right].
\end{aligned}$$

On the other hand, notice that

$$\left(\bar{\sigma}_t^{\text{VSPO}}\right)^2=\frac{1}{G-1}\left(\sum_{g=1}^m r_t^+\left(a_{t,+}^{(g)}\right)^2+\sum_{g=1}^m r_t^-\left(a_{t,-}^{(g)}\right)^2-G(\bar{r}_t^{\text{VSPO}})^2\right).$$

Recall the reward means

$$\mu_t^+:=\sum_{j=1}^K\pi_t^+(j)r_+(j)=\mu_{x,t}^++\alpha\mu_{y,t}^+,\ \mu_t^-:=\sum_{j=1}^K\pi_t^-(j)r_-(j)=\mu_{x,t}^-+\alpha\mu_{y,t}^-,$$

and

$$\mu_t:=\frac{\mu_t^++\mu_t^-}{2}=\mu_{x,t}+\alpha\mu_{y,t},\ \delta_t:=\mu_t^+-\mu_t^-=\delta_{x,t}+\alpha\delta_{y,t}.$$

Define the within-group variances

$$\begin{aligned}
\left(\sigma_t^+\right)^2&:=\sum_i\pi_t^+(i)\left(r_t^+(i)-\mu_t^+\right)^2=\sum_i\pi_t^+(i)\left(x(i)-\mu_{x,t}^+\right)^2,\\
\left(\sigma_t^-\right)^2&:=\sum_i\pi_t^-(i)\left(r_t^-(i)-\mu_t^-\right)^2=\sum_i\pi_t^-(i)\left(x(i)-\mu_{x,t}^-\right)^2,
\end{aligned}$$

and also recall the pooled population variance

$$v_{r,t}^2:=\frac{1}{2}\sum_i\pi_t^+(i)\left(r_t^+(i)-\mu_t\right)^2+\frac{1}{2}\sum_i\pi_t^-(i)\left(r_t^-(i)-\mu_t\right)^2.$$

We first relate $v_{r,t}^2$ to $\left(\sigma_t^+\right)^2,\left(\sigma_t^-\right)^2$. By expanding around μ_t, we have

$$\begin{aligned}
v_{r,t}^2&=\frac{1}{2}\sum_i\pi_t^+(i)\left(r_t^+(i)-\mu_t\right)^2+\frac{1}{2}\sum_i\pi_t^-(i)\left(r_t^-(i)-\mu_t\right)^2\\
&=\frac{1}{2}\left[\left(\sigma_t^+\right)^2+(\mu_t^+-\mu_t)^2\right]+\frac{1}{2}\left[\left(\sigma_t^-\right)^2+(\mu_t^--\mu_t)^2\right]\\
&=\frac{\left(\sigma_t^+\right)^2+\left(\sigma_t^-\right)^2}{2}+\frac{\delta_t^2}{4},
\end{aligned}$$

which implies

$$\frac{\left(\sigma_t^+\right)^2+\left(\sigma_t^-\right)^2}{2}=v_{r,t}^2-\frac{\delta_t^2}{4}. \tag{21}$$

Similarly, we have

$$\mu_t^2=\left(\frac{\mu_t^++\mu_t^-}{2}\right)^2=\frac{\left(\mu_t^+\right)^2+\left(\mu_t^-\right)^2}{2}-\frac{(\mu_t^+-\mu_t^-)^2}{4}=\frac{\left(\mu_t^+\right)^2+\left(\mu_t^-\right)^2}{2}-\frac{\delta_t^2}{4},$$

which implies

$$\frac{\left(\mu_t^+\right)^2+\left(\mu_t^-\right)^2}{2}=\mu_t^2+\frac{\delta_t^2}{4}.$$

Thus we have

$$\begin{aligned}
&\frac{1}{G}\mathbb{E}\left[\sum_{g=1}^{m} r_t^+ \left(a_{t,+}^{(g)}\right)^2 + \sum_{g=1}^{m} r_t^- \left(a_{t,-}^{(g)}\right)^2 \,\middle|\, \pi_t^+, \pi_t^-\right] \\
&= \frac{1}{2}\left(\left(\sigma_t^+\right)^2 + (\mu_t^+)^2\right) + \frac{1}{2}\left(\left(\sigma_t^-\right)^2 + (\mu_t^-)^2\right) \\
&= \frac{\left(\sigma_t^+\right)^2 + \left(\sigma_t^-\right)^2}{2} + \frac{\left(\mu_t^+\right)^2 + \left(\mu_t^-\right)^2}{2} \\
&= v_{r,t}^2 + \mu_t^2
\end{aligned}$$

Notice that

$$\mathbb{E}[\bar{r}_t^{\text{VSPO}} \mid \pi_t^+, \pi_t^-] = \mu_t,$$

and by independence,

$$\begin{aligned}
\text{Var}(\bar{r}_t^{\text{VSPO}} \mid \pi_t^+, \pi_t^-) &= \frac{1}{G^2}\left(m\left(\sigma_t^+\right)^2 + m\left(\sigma_t^-\right)^2\right) \\
&= \frac{\left(\sigma_t^+\right)^2 + \left(\sigma_t^-\right)^2}{2G} = \frac{1}{G}\left(v_{r,t}^2 - \frac{\delta_t^2}{4}\right),
\end{aligned}$$

where the last step uses Eqn. (21). Therefore,

$$\mathbb{E}[\left(\bar{r}_t^{\text{VSPO}}\right)^2 \mid \pi_t^+, \pi_t^-] = \mu_t^2 + \frac{1}{G}\left(v_{r,t}^2 - \frac{\delta_t^2}{4}\right).$$

Substituting into the definition of $\bar{\sigma}_t^2$, we obtain

$$\begin{aligned}
\mathbb{E}[\left(\bar{\sigma}_t^{\text{VSPO}}\right)^2 \mid \pi_t^+, \pi_t^-] &= \frac{G}{G-1}\left(v_{r,t}^2 + \mu_t^2\right) - \frac{G}{G-1}\left[\mu_t^2 + \frac{1}{G}\left(v_{r,t}^2 - \frac{\delta_t^2}{4}\right)\right] \\
&= v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)},
\end{aligned}$$

as desired. □

Remark 9. In Propositions 2 and 4, we use the VSPO population arm-level score as

$$A_t^{\text{VSPO}}(i) := \frac{(G-1)\,\bar{\pi}_t(i)\big(x(i) - \mu_{x,t}\big) + \frac{d_t(i)}{2}\left(\delta_t + \alpha(G-1)\delta_{y,t}\right)}{\sqrt{v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}}}. \tag{22}$$

E.4 Technical Lemmas

Lemma 10 (Soft policy update)**.** *For any arm-level advantage score $A_t \in \mathbb{R}^K$, the unique optimizer of Eqn.* (6) *is*

$$\pi_{t+1}(i) = \frac{\pi_t(i)\exp\left(\frac{\eta A_t(i)}{G\pi_t(i)}\right)}{\sum_{j=1}^{K}\pi_t(j)\exp\left(\frac{\eta A_t(j)}{G\pi_t(j)}\right)}. \tag{23}$$

Proof. The objective in Eqn. (6) is

$$\pi_{t+1} = \arg\max_{\pi}\left\{\frac{1}{G}\sum_{i=1}^{K}\frac{\pi(i)}{\pi_t(i)}A_t(i) - \frac{1}{\eta}\sum_{i=1}^{K}\pi(i)\log\frac{\pi(i)}{\pi_t(i)}\right\},$$

which is strictly concave over the simplex, so it has a unique maximizer. Introduce the Lagrangian

$$\mathcal{L}(\pi, \lambda) = \sum_{i=1}^{K}\pi(i)\frac{A_t(i)}{G\pi_t(i)} - \frac{1}{\eta}\sum_{i=1}^{K}\pi(i)\log\frac{\pi(i)}{\pi_t(i)} + \lambda\left(\sum_{i=1}^{K}\pi(i) - 1\right).$$

Taking derivatives with respect to $\pi(i)$ gives

$$0 = \frac{\partial \mathcal{L}}{\partial \pi(i)} = \frac{A_t(i)}{G\pi_t(i)} - \frac{1}{\eta}\left(\log \frac{\pi(i)}{\pi_t(i)} + 1\right) + \lambda.$$

Rearranging,

$$\pi(i) = \pi_t(i) \exp\left(\frac{\eta A_t(i)}{G\pi_t(i)} + \eta\lambda - 1\right).$$

The term $\exp(\eta\lambda - 1)$ is the same for every arm and is determined by the normalization constraint $\sum_i \pi(i) = 1$. Substituting the normalizing constant yields Eqn. (23). □

Lemma 11. *For any bounded random variable $X \in [a, b]$, we have $\mathrm{Var}(X) \le \frac{(b-a)^2}{4}$.*

Proof. Notice that $(X - a)(X - b) \le 0$, which implies $X^2 \le (a + b)X - ab$. Therefore, we have

$$\begin{aligned}\mathrm{Var}(X) &= \mathbb{E}[X^2] - \mathbb{E}^2[X] \le (a + b)\mathbb{E}[X] - ab - \mathbb{E}^2[X] \\ &= -\left(\mathbb{E}[X] - \frac{a+b}{2}\right)^2 + \frac{(b-a)^2}{4} \le \frac{(b-a)^2}{4},\end{aligned}$$

as desired. □

E.5 Proof of Proposition 1

Proposition 1 (Iteration complexity of GRPO)**.** *Under the GRPO population score A_t^{GRPO} in Appendix E.2, we have $J(\pi_T) \ge r^\star - \varepsilon$ when*

$$T \ge T^{\mathrm{GRPO}}(\varepsilon) := \frac{\Delta_{\max}}{2\eta\left(1 - \frac{1}{G}\right)\Delta_{\min}} \log \frac{C_0}{\varepsilon}.$$

Proof. For every suboptimal arm $i \ne i^\star$, define

$$q_{i,t} := \frac{\pi_t(i)}{\pi_t(i^\star)}.$$

By Lemma 10 and Eqn. (20), we have

$$\frac{\pi_{t+1}(i)}{\pi_{t+1}(i^\star)} = \frac{\pi_t(i)}{\pi_t(i^\star)} \exp\left[\eta\left(\frac{A_t^{\mathrm{RS}}(i)}{G\pi_t(i)} - \frac{A_t^{\mathrm{RS}}(i^*)}{G\pi_t(i^\star)}\right)\right] = \frac{\pi_t(i)}{\pi_t(i^\star)} \exp\left[\eta\left(1 - \frac{1}{G}\right)\frac{r(i) - r(i^\star)}{\sigma_{r,t}}\right],$$

which implies

$$q_{i,t+1} = q_{i,t} \exp\left[-\eta\left(1 - \frac{1}{G}\right)\frac{\Delta_i}{\sigma_{r,t}}\right] \le q_{i,t} \exp\left[-\eta\left(1 - \frac{1}{G}\right)\frac{\Delta_{\min}}{\sigma_{r,t}}\right].$$

Since r is bounded by $[r(i^\star) - \Delta_{\max}, r(i^\star)]$, by Lemma 11 we have $\sigma_{r,t} \le \frac{\Delta_{\max}}{2}$. This implies

$$q_{i,t+1} \le q_{i,t} \exp\left[-2\eta\left(1 - \frac{1}{G}\right)\frac{\Delta_{\min}}{\Delta_{\max}}\right].$$

Iterating gives

$$q_{i,t} \le q_{i,0} \exp\left(-\frac{2\eta\left(1 - \frac{1}{G}\right)\Delta_{\min}}{\Delta_{\max}} t\right).$$

Notice that

$$r(i^\star) - J(\pi_t) = \sum_{i \ne i^\star} \pi_t(i)\Delta_i = \pi_t(i^\star) \sum_{i \ne i^\star} q_{i,t}\Delta_i \le \sum_{i \ne i^\star} q_{i,t}\Delta_i,$$

which implies

$$\begin{aligned}r(i^\star) - J(\pi_t) &\le \sum_{i \ne i^\star} q_{i,t}\Delta_i \le \exp\left(-\frac{2\eta\left(1 - \frac{1}{G}\right)\Delta_{\min}}{\Delta_{\max}} t\right) \sum_{i \ne i^\star} q_{i,0}\Delta_i \\ &= C_0 \exp\left(-\frac{2\eta\left(1 - \frac{1}{G}\right)\Delta_{\min}}{\Delta_{\max}} t\right).\end{aligned}$$

This further implies when

$$T \geq T^{\text{GRPO}}(\varepsilon) = \frac{\Delta_{\max}}{2\eta\left(1-\frac{1}{G}\right)\Delta_{\min}} \log \frac{C_0}{\varepsilon},$$

we have

$$r(i^\star) - J(\pi_T) \leq C_0 \exp\left(-\frac{2\eta\left(1-\frac{1}{G}\right)\Delta_{\min}}{\Delta_{\max}} T^{\text{GRPO}}(\epsilon)\right) \leq \epsilon,$$

as desired. □

E.6 Proof of Proposition 2

Proposition 2 (Iteration complexity of VSPO). *Define distributional reward gaps*

$$\delta_{x,t} := \mathbb{E}_{i\sim\pi_t^+}[x(i)] - \mathbb{E}_{i\sim\pi_t^-}[x(i)],\ \delta_{y,t} := \mathbb{E}_{i\sim\pi_t^+}[y(i)] - \mathbb{E}_{i\sim\pi_t^-}[y(i)],\ \delta_t := \delta_{x,t} + \alpha\delta_{y,t},$$

distributional gap $d_t(i) := \frac{\pi_t^+(i)-\pi_t^-(i)}{2}$, *and relative distributional contrast* $\rho_t(i) := \frac{d_t(i)}{\pi_t(i)}$. *Assume* $\bar{\pi}_t = \pi_t$ *for all* $t \geq 0$, *and the vector steering is* γ*-good: for all* $t \geq 0$ *and all* $i \neq i^\star$,

$$\frac{\delta_t}{2G}\left(\rho_t(i^\star) - \rho_t(i)\right) \geq \gamma,\ \delta_{y,t}\left(\rho_t(i^\star) - \rho_t(i)\right) \geq 2\left(y(i^\star) - y(i)\right).$$

Then, under the VSPO population score A_t^{VSPO} *in Appendix E.3, we have* $J(\pi_T) \geq r(i^\star) - \varepsilon$ *when*

$$T \geq T^{\text{VSPO}}(\varepsilon) := \frac{\lambda\Delta_{\max}}{2\eta\sqrt{1-\frac{1}{G}}\left(\left(1-\frac{1}{G}\right)\Delta_{\min} + \gamma\right)} \log \frac{C_0}{\varepsilon},$$

where λ *is the bandit conditioning constant defined in Eqn. (3).*

Proof. For every suboptimal arm $i \neq i^\star$, define

$$q_{i,t} := \frac{\pi_t(i)}{\pi_t(i^\star)}.$$

By Lemma 10 and Eqn. (22), we have

$$\begin{aligned}
&\sqrt{v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}}\left(\frac{A_t^{\text{VSPO}}(i)}{G\pi_t(i)} - \frac{A_t^{\text{VSPO}}(i^*)}{G\pi_t(i^\star)}\right) \\
&= \left(1-\frac{1}{G}\right)(x(i) - x(i^\star)) - \frac{\delta_t + \alpha(G-1)\delta_{y,t}}{2G}\left(\frac{d_t(i^\star)}{\pi_t(i^\star)} - \frac{d_t(i)}{\pi_t(i)}\right) \\
&\leq \left(1-\frac{1}{G}\right)(x(i) - x(i^\star)) - \gamma - \frac{\alpha(G-1)}{2G}\cdot 2(y(i^\star) - y(i)) \\
&= \left(1-\frac{1}{G}\right)(r(i) - r(i^\star)) - \gamma \\
&\leq -\left(1-\frac{1}{G}\right)\Delta_{\min} - \gamma.
\end{aligned}$$

Also, since $r_t^+(i), r_t^-(i)$ is bounded by

$$\left[\min_{i\in[K]} x(i) + \alpha \min_{i\in[K]} y(i), \max_{i\in[K]} x(i) + \alpha \max_{i\in[K]} y(i)\right],$$

by Lemma 11 we have $\sigma_t \leq \frac{D_x + \alpha D_y}{2} = \frac{\lambda\Delta_{\max}}{2}$. Furthermore, since $\delta_t = \mu_t^+ - \mu_t^-$, we have $|\delta_t| \leq D_x + \alpha D_y = \lambda\Delta_{\max}$. Thus we have

$$\sqrt{v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}} \leq \sqrt{\frac{(\lambda\Delta_{\max})^2}{4} + \frac{(\lambda\Delta_{\max})^2}{4(G-1)}} = \frac{\lambda\Delta_{\max}}{2\sqrt{1-\frac{1}{G}}}.$$

Combining the inequalities above yields

$$\frac{A_t^{\text{VSPO}}(i)}{G\pi_t(i)} - \frac{A_t^{\text{VSPO}}(i^*)}{G\pi_t(i^\star)} \le -\frac{\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma}{\sqrt{v_{r,t}^2+\frac{\delta_t^2}{4(G-1)}}} \le -\frac{2\sqrt{1-\frac{1}{G}}}{\lambda\Delta_{\max}}\left[\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right],$$

which implies

$$\begin{aligned} q_{i,t+1} = \frac{\pi_{t+1}(i)}{\pi_{t+1}(i^\star)} &= \frac{\pi_t(i)}{\pi_t(i^\star)}\exp\left[\eta\left(\frac{A_t^{\text{VSPO}}(i)}{G\pi_t(i)} - \frac{A_t^{\text{VSPO}}(i^*)}{G\pi_t(i^\star)}\right)\right] \\ &\le q_{i,t}\exp\left\{-\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda\Delta_{\max}}\left[\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right]\right\}. \end{aligned}$$

Iterating gives

$$q_{i,t} \le q_{i,0}\exp\left\{-\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda\Delta_{\max}}\left[\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right]t\right\}.$$

Notice that

$$r(i^\star) - J(\pi_t) = \sum_{i\neq i^\star}\pi_t(i)\Delta_i = \pi_t(i^\star)\sum_{i\neq i^\star} q_{i,t}\Delta_i \le \sum_{i\neq i^\star} q_{i,t}\Delta_i,$$

which implies

$$\begin{aligned} r(i^\star) - J(\pi_t) &\le \sum_{i\neq i^\star} q_{i,t}\Delta_i \le \exp\left\{-\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda\Delta_{\max}}\left[\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right]t\right\}\sum_{i\neq i^\star} q_{i,0}\Delta_i \\ &= C_0\exp\left\{-\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda\Delta_{\max}}\left[\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right]t\right\}. \end{aligned}$$

This further implies when

$$T \ge T^{\text{VSPO}}(\varepsilon) = \frac{\lambda\Delta_{\max}}{2\eta\sqrt{1-\frac{1}{G}}\left(\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right)}\log\frac{C_0}{\varepsilon},$$

we have

$$r(i^\star) - J(\pi_T) \le C_0\exp\left\{-\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda\Delta_{\max}}\left[\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right]T^{\text{VSPO}}(\varepsilon)\right\} \le \varepsilon,$$

as desired. □

E.7 Proof of Corollary 3

Corollary 3. *When γ defined in Eqn. (8) satisfies $\gamma > \min\left\{\lambda, \frac{\lambda^2}{4}\right\}\Delta_{\min}$, then $T^{\text{VSPO}}(\varepsilon) < T^{\text{GRPO}}(\varepsilon)$, where $T^{\text{GRPO}}(\varepsilon), T^{\text{VSPO}}(\varepsilon)$ are defined in Eqns. (7) and (9) respectively.*

Proof. Notice that

$$\left[\lambda\sqrt{1-\frac{1}{G}} - \left(1-\frac{1}{G}\right)\right]\Delta_{\min} < \lambda\Delta_{\min}, \quad \left[\lambda\sqrt{1-\frac{1}{G}} - \left(1-\frac{1}{G}\right)\right]\Delta_{\min} \le \frac{\lambda^2}{4}\Delta_{\min},$$

which implies

$$\lambda > \min\left\{\lambda, \frac{\lambda^2}{4}\right\}\Delta_{\min} \ge \left[\lambda\sqrt{1-\frac{1}{G}} - \left(1-\frac{1}{G}\right)\right]\Delta_{\min}.$$

Thus we have

$$T^{\text{VSPO}}(\varepsilon) = \frac{\lambda\Delta_{\max}}{2\eta\sqrt{1-\frac{1}{G}}\left(\left(1-\frac{1}{G}\right)\Delta_{\min}+\gamma\right)} < \frac{\lambda\Delta_{\max}}{2\eta\sqrt{1-\frac{1}{G}}\cdot\lambda\sqrt{1-\frac{1}{G}}\Delta_{\min}} = T^{\text{GRPO}}(\varepsilon),$$

as desired. □

E.8 More on γ-Good Condition

Lemma 12. *$\gamma \le \frac{\lambda \Delta_{\max}}{G}$ where γ is defined in Eqn. (8), and λ is the bandit conditioning constant defined in Eqn. (3).*

Proof. Notice that for any arm i, we have

$$d_t(i) = \frac{\pi_t^+(i) - \pi_t^-(i)}{2},\ \pi_t(i) = \frac{\pi_t^+(i) + \pi_t^-(i)}{2},$$

which implies $\pi_t^+(i) = \pi_t(i) + d_t(i)$. Thus we have

$$\pi_t^+(i) = \pi_t(i) + d_t(i) \le \pi_t(i) + \frac{\pi_t^+(i)}{2} \implies \pi_t^+(i) \le 2\pi_t(i),$$

which implies

$$\rho_t(i) = \frac{d_t(i)}{\pi_t(i)} = \frac{\pi_t^+(i) - \pi_t(i)}{\pi_t(i)} = \frac{\pi_t^+(i)}{\pi_t(i)} - 1 \in [-1, 1].$$

Thus we have $-2 \le \rho_t(i^\star) - \rho_t(i) \le 2$ for any arm i.

Since $\delta_t = \mu_t^+ - \mu_t^-$, we have $|\delta_t| \le D_x + \alpha D_y = \lambda \Delta_{\max}$. This implies

$$\gamma \le \frac{\delta_t}{2G}\left(\rho(i^\star) - \rho(i)\right) \le \frac{|\delta_t|}{2G}\left|\rho(i^\star) - \rho(i)\right| \le \frac{\lambda \Delta_{\max}}{G},$$

as desired. □

E.9 Proof of Proposition 4

Proposition 4. *Assume $\max_{t,i}\left|\frac{\bar{\pi}_t(i)}{\pi_t(i)} - 1\right| \le \kappa$, and the vector steering is γ-good as defined in Proposition 2. Then, under the VSPO population score A_t^{VSPO} in Appendix E.3, we have $J(\pi_T) \ge r(i^\star) - \varepsilon$ when*

$$T \ge \frac{\lambda \Delta_{\max}}{2\eta\sqrt{1 - \frac{1}{G}}\left(\left(1 - \frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1 - \frac{1}{G}\right)\kappa D_x\right)} \log \frac{C_0}{\varepsilon}, \tag{24}$$

where λ, D_x are defined in Eqn. (3) and assuming $\left(1 - \frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1 - \frac{1}{G}\right)\kappa D_x > 0$.

Proof. For every suboptimal arm $i \ne i^\star$, define

$$q_{i,t} := \frac{\pi_t(i)}{\pi_t(i^\star)}.$$

By Lemma 10 and Eqn. (22), we have

$$
\begin{aligned}
&\sqrt{v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}} \left(\frac{A_t^{\text{VSPO}}(i)}{G\pi_t(i)} - \frac{A_t^{\text{VSPO}}(i^*)}{G\pi_t(i^\star)} \right) \\
&= \left(1 - \frac{1}{G}\right) \left[\frac{\bar{\pi}_t(i)}{\pi_t(i)} (x(i) - \mu_{x,t}) - \frac{\bar{\pi}_t(i^\star)}{\pi_t(i^\star)} (x(i^\star) - \mu_{x,t}) \right] \\
&\quad - \frac{\delta_t + \alpha(G-1)\delta_{y,t}}{2G} \left(\frac{d_t(i^\star)}{\pi_t(i^\star)} - \frac{d_t(i)}{\pi_t(i)} \right) \\
&= \left(1 - \frac{1}{G}\right) \left[(x(i) - x(i^\star)) + \left(\frac{\bar{\pi}_t(i)}{\pi_t(i)} - 1 \right) (x(i) - \mu_{x,t}) - \left(\frac{\bar{\pi}_t(i^\star)}{\pi_t(i^\star)} - 1 \right) (x(i^\star) - \mu_{x,t}) \right] \\
&\quad - \frac{\delta_t + \alpha(G-1)\delta_{y,t}}{2G} \left(\frac{d_t(i^\star)}{\pi_t(i^\star)} - \frac{d_t(i)}{\pi_t(i)} \right) \\
&\le \left(1 - \frac{1}{G}\right) (x(i) - x(i^\star)) + 2\left(1 - \frac{1}{G}\right) \kappa D_x - \gamma - \frac{\alpha(G-1)}{2G} \cdot 2(y(i^\star) - y(i)) \\
&= \left(1 - \frac{1}{G}\right) (r(i) - r(i^\star)) - \gamma + 2\left(1 - \frac{1}{G}\right) \kappa D_x \\
&\le -\left(1 - \frac{1}{G}\right) \Delta_{\min} - \gamma + 2\left(1 - \frac{1}{G}\right) \kappa D_x.
\end{aligned}
$$

Also, since $r_t^+(i), r_t^-(i)$ is bounded by

$$
\left[\min_{i \in [K]} x(i) + \alpha \min_{i \in [K]} y(i), \max_{i \in [K]} x(i) + \alpha \max_{i \in [K]} y(i) \right],
$$

by Lemma 11 we have $\sigma_t \le \frac{D_x + \alpha D_y}{2} = \frac{\lambda \Delta_{\max}}{2}$. Furthermore, since $\delta_t = \mu_t^+ - \mu_t^-$, we have $|\delta_t| \le D_x + \alpha D_y = \lambda \Delta_{\max}$. Thus we have

$$
\sqrt{v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}} \le \sqrt{\frac{(\lambda \Delta_{\max})^2}{4} + \frac{(\lambda \Delta_{\max})^2}{4(G-1)}} = \frac{\lambda \Delta_{\max}}{2\sqrt{1 - \frac{1}{G}}}.
$$

Combining the inequalities above yields

$$
\begin{aligned}
\frac{A_t^{\text{VSPO}}(i)}{G\pi_t(i)} - \frac{A_t^{\text{VSPO}}(i^*)}{G\pi_t(i^\star)} &\le -\frac{\left(1 - \frac{1}{G}\right) \Delta_{\min} + \gamma - 2\left(1 - \frac{1}{G}\right) \kappa D_x}{\sqrt{v_{r,t}^2 + \frac{\delta_t^2}{4(G-1)}}} \\
&\le -\frac{2\sqrt{1 - \frac{1}{G}}}{\lambda \Delta_{\max}} \left[\left(1 - \frac{1}{G}\right) \Delta_{\min} + \gamma - 2\left(1 - \frac{1}{G}\right) \kappa D_x \right],
\end{aligned}
$$

which implies

$$
\begin{aligned}
q_{i,t+1} = \frac{\pi_{t+1}(i)}{\pi_{t+1}(i^\star)} &= \frac{\pi_t(i)}{\pi_t(i^\star)} \exp\left[\eta \left(\frac{A_t^{\text{VSPO}}(i)}{G\pi_t(i)} - \frac{A_t^{\text{VSPO}}(i^*)}{G\pi_t(i^\star)} \right) \right] \\
&\le q_{i,t} \exp\left\{ -\frac{2\eta\sqrt{1 - \frac{1}{G}}}{\lambda \Delta_{\max}} \left[\left(1 - \frac{1}{G}\right) \Delta_{\min} + \gamma - 2\left(1 - \frac{1}{G}\right) \kappa D_x \right] \right\}.
\end{aligned}
$$

Iterating gives

$$
q_{i,t} \le q_{i,0} \exp\left\{ -\frac{2\eta\sqrt{1 - \frac{1}{G}}}{\lambda \Delta_{\max}} \left[\left(1 - \frac{1}{G}\right) \Delta_{\min} + \gamma - 2\left(1 - \frac{1}{G}\right) \kappa D_x \right] t \right\}.
$$

Notice that

$$
r(i^\star) - J(\pi_t) = \sum_{i \ne i^\star} \pi_t(i) \Delta_i = \pi_t(i^\star) \sum_{i \ne i^\star} q_{i,t} \Delta_i \le \sum_{i \ne i^\star} q_{i,t} \Delta_i,
$$

which implies

$$
\begin{aligned}
r(i^\star) - J(\pi_t) &\le \sum_{i \neq i^\star} q_{i,t} \Delta_i \\
&\le \exp\left\{ -\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda \Delta_{\max}} \left[\left(1-\frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1-\frac{1}{G}\right)\kappa D_x \right] t \right\} \sum_{i \neq i^\star} q_{i,0}\Delta_i \\
&= C_0 \exp\left\{ -\frac{2\eta\sqrt{1-\frac{1}{G}}}{\lambda \Delta_{\max}} \left[\left(1-\frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1-\frac{1}{G}\right)\kappa D_x \right] t \right\}.
\end{aligned}
$$

This further implies when

$$
T \ge \frac{\lambda \Delta_{\max}}{2\eta\sqrt{1-\frac{1}{G}}\left(\left(1-\frac{1}{G}\right)\Delta_{\min} + \gamma - 2\left(1-\frac{1}{G}\right)\kappa D_x\right)} \log \frac{C_0}{\varepsilon},
$$

we have $r(i^\star) - J(\pi_T) \le \varepsilon$, as desired. □

F Baseline

We compare VSPO against representative approaches that can be adapted to learning controllable reasoning behaviors. These baselines cover teacher-supervised distillation, self-distillation, prompt-conditioned RL, and reward-shaped RL. Importantly, several methods, such as SDFT and SDPO, were not originally designed for controllable model behavior. They primarily aim to improve task performance through self-distillation. To make them comparable in our setting, we adapt their selection, prompting, or reward signals to include the target behavioral attribute when applicable.

- **Teacher-Trace Distillation (SFT).** We adapt recent short-CoT distillation methods, which train models on compressed reasoning traces to reduce inference length [8, 13], to our broader controllable-behavior settings. Specifically, we use an external teacher model to generate reasoning traces aligned with each target behavior, such as expert-level explanations and concise reasoning, and then fine-tune the student model with Supervised Fine-Tuning (SFT). This baseline follows the standard offline teacher-distillation paradigm [16]. For fair comparison, we use the same number of trace as our vector construction.
- **Teacher-Trace Distillation (SFT) + GRPO.** We further evaluate a two-stage variant that first performs teacher-trace SFT and then applies GRPO. This follows the standard post-training recipe in which SFT provides an initial policy aligned with the desired response format or behavior, and RL further optimizes the model for task reward [6, 32]. For this baseline, we use the Teacher-Trace Distillation (SFT) checkpoint as the initialization and continue training with GRPO for the same number of RL steps as VSPO, namely 100 steps.
- **SDFT (Self-Distillation Fine-Tuning).** We include Self-Distillation Fine-Tuning (SDFT) [21] as a self-distillation baseline. SDFT uses the model itself as a teacher under demonstration-conditioned prompting, producing training signals from the model's own conditional distribution rather than from fixed external traces. Since SDFT was originally proposed for improving policy learning rather than controllable model behavior, we adapt it by conditioning demonstrations and target behavior prompts on the target attribute.
- **SDPO (Self-Distillation Policy Optimization).** We compare with Self-Distillation Policy Optimization (SDPO) [10], a self-distillation RL method that improves the policy using its own high-quality rollouts. The model samples multiple candidate responses and treats stronger outputs (e.g., based on correctness and our target) as implicit targets, which are then optimized via a GRPO-style objective. In its original form, SDPO is designed for policy improvement, not for controllable model behavior. We therefore adapt it to our setting by selecting stronger rollouts according to a combined criterion for both correctness and the target attribute. This gives SDPO access to the same behavioral objective used in our evaluation. We train 100 steps for fair comparison.
- **Reward-Shaped GRPO.** We incorporate the target behavior directly into the GRPO reward. For concise reasoning, we add a length penalty to encourage shorter responses [23, 1, 32]. For Preference-Aligned Reasoning Behavior, such as expertise-level control, we use an LLM-based evaluator to reward responses that match the desired user preference. Because this baseline places

an LLM inside the RL loop for evaluation, it incurs substantial computational and monetary cost. For Robustness to Misleading Context, we do not introduce an additional behavior-specific reward, since the desired behavior is reflected by task accuracy. We train 100 steps for fair comparison.

- **GRPO with Off-Policy Textual Guidance.** We include a text-guided GRPO baseline inspired by recent RL methods that augment standard rollouts with auxiliary textual supervision, such as instructions or feedback [30, 27, 22, 29]. These methods use additional context to elicit desired behaviors during generation and then train the model to internalize those behaviors so they can be produced without the extra context at inference time. Concretely, we append the target-behavior instruction to the original prompt only during rollout generation. During optimization, we remove this instruction and compute policy log probabilities under the original prompt, updating the unprompted policy with the standard GRPO objective [19]. This baseline tests whether our latent vector steering provides a more effective mechanism for eliciting optimization-useful target-behavior rollouts than explicit behavior-inducing prompts.

We do not adopt the finetuning-based intervention from Persona Vectors [5] because its goal is different from ours. Persona Vectors studies how latent trait directions can monitor, predict, and prevent undesirable persona shifts during finetuning, including a preventative steering procedure designed to suppress harmful personality drift. In contrast, our goal is not to prevent an unwanted trait from emerging during standard finetuning, but to use a target behavior direction as an exploration mechanism inside RL. VSPO therefore applies vector steering only during rollout generation, evaluates the resulting behavior-accuracy trade-off with GRPO rewards, and updates the unsteered policy. The results of the finetuning-based intervention from Persona Vectors are reported in Table 4. This approach performs substantially worse than the baselines considered in the main paper, both in expertise-level control and task accuracy, so we do not include it as a primary baseline.

G Experimental details

Compute Resources and Reproducibility. We report the compute resources used for each stage of the experimental pipeline. Claude API calls for teacher-assisted rewriting and LLM-based behavioral evaluation are performed through AWS Bedrock. Steering-vector construction consists of forward passes over contrastive response pairs and is run on $4\times$A40 GPUs. Policy optimization experiments are run on GPU nodes with either $8\times$H200 GPUs or 4–$8\times$L40S GPUs, depending on the model and training configuration. Each training run uses the same GPU type within a run. Unless otherwise specified, all inference, and downstream evaluation are performed on $4\times$A40 GPUs. Unless otherwise specified, we train 100 steps for RL and 2 epochs for SFT.

The main experiments use 4-bit or bfloat16 model inference depending on the backend configuration, and all policy optimization runs use distributed training with vLLM-based rollout generation. Wall-clock time varies with model size and rollout length.

Comparison protocol. The GRPO-family comparisons use the same base model, batch size, number of generated responses per update, context window, training steps, and shared GRPO hyperparameters. We use officially recommended method-specific settings when available; otherwise, we tune method-specific hyperparameters while keeping shared optimization settings fixed. VSPO uses the recommended VERL defaults for shared optimization settings, including the learning rate. These comparisons match rollout and update counts; total computational cost also depends on response lengths and external-model calls.

G.1 Vector steering

G.1.1 Steering Vector Construction.

We construct steering vectors using a teacher-assisted contrastive pipeline. For each target model and target property, we first sample reasoning traces from the target model itself (e.g., Qwen3-4B) and retain only traces that lead to the correct answer, allowing up to five sampling attempts per prompt. Each retained trace is then rewritten by Claude in two opposite directions of the target property, yielding a positive and a negative trace pair (e.g., expert vs. elementary, or confident vs. cautious). We again allow up to five rewriting attempts and filter the rewritten traces by both task correctness and a property-specific score threshold; the thresholds for each setting are reported in Table 3 in the appendix. In total, we collect 200 positive–negative rewrite pairs for each target property.

Property	Rewrite Pair	Positive Condition	Negative Condition
Expertise Level	Expert / Elementary	≥ 70	≤ 30
Confidence Expression	Confident / Cautious	≥ 60	≤ 40
Robustness to Misleading Context	Robust / Sycophantic	Correct answer	Incorrect answer
Concise Reasoning	Concise / Verbose	≥ 1536	≥ 512

Table 3: Property-specific filtering thresholds for teacher-generated contrastive rewrites. A rewrite pair is retained only if both rewrites satisfy the corresponding filtering conditions. For expertise level and confidence expression, we use a 0–100 evaluator score and retain pairs whose positive rewrite scores at least 70 and whose negative rewrite scores at most 20. For robustness to misleading context, the contrast is defined by answer correctness under the misleading prompt rather than by a continuous style score. For concise reasoning, lower verbosity indicates stronger alignment with the concise direction.

Method	Expertise Level ↑	Accuracy ↑
Teacher-Trace Distillation (SFT)	45.0	66.3
Finetuning-Based Intervention from Persona Vectors	30.7	44.8
VSPO (Ours)	**64.3**	**75.4**

Table 4: Comparison with the finetuning-based intervention from Persona Vectors on MMLU-Pro under the expert target behavior. VSPO achieves substantially stronger expert-style control while also improving task accuracy.

Following [5], we compute a candidate steering vector at each layer as the mean activation difference between the positive and negative rewritten traces, conditioned on the same original prompt:

$$v_\ell = \frac{1}{N} \sum_{i=1}^{N} \left(h_\ell^+(x_i, y_i^+) - h_\ell^-(x_i, y_i^-) \right),$$

where h_ℓ^+ and h_ℓ^- denote the layer-ℓ activations induced by the positive and negative traces, respectively. We then select the layer whose vector induces the strongest separation along the target-property evaluator. To make steering coefficients comparable across target properties and tasks, we normalize the selected vector according to its empirical scoring magnitude, so that a unit steering coefficient approximately corresponds to moving the generation toward the maximum or minimum end of the observed property-score range. The resulting vector is fixed throughout RL optimization and is only used during rollout generation.

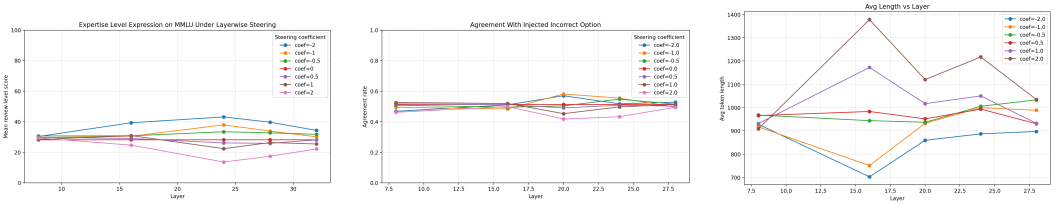
In practice, we recommend using smaller β values since RL iterations can always push to the desired direction gradually, and note that the vector is fixed throughout optimization.

To construct each steering vector, we compute contrastive activation differences between positive and negative behavior demonstrations at each transformer layer. Since the separability and controllability of the target behavior can vary substantially across layers, we perform a layer search by applying vectors extracted from different layers and evaluating the resulting accuracy–behavior trade-off. As shown in Figure 8, the best layer is selected separately for each target behavior and then fixed for all subsequent vector-steered rollouts.

G.1.2 Expertise level

Elementary-Level Explanation: The elementary-level condition is designed to elicit solutions written for readers with minimal mathematical background. Responses in this style use simple language, short sentences, and explicit step-by-step reasoning. The model is instructed to avoid advanced terminology and dense notation whenever possible, to explain the meaning of each quantity or operation, and to present calculations slowly and transparently. The target style is pedagogical and beginner-oriented: routine steps are not skipped, symbols are introduced sparingly, and the solution is structured to make the reasoning accessible to a young or novice learner.

Expert-Level Explanation: The expert-level condition is designed to elicit solutions written for mathematically sophisticated readers. Responses in this style use precise mathematical language,



(a) Expertise level (b) Misleading Context (c) Concise reasoning

Figure 8: Layer selection for steering-vector construction. For each target behavior, we evaluate steering vectors computed from different transformer layers and select the layer that best improves the target behavioral score while preserving task accuracy. The selected layer is then fixed and used for all subsequent vector-steered rollouts in VSPO.

standard notation, and compact derivations. The model is instructed to omit routine intermediate steps, compress algebraic or arithmetic manipulations, and avoid pedagogical explanations of basic concepts. The target style is concise, formal, and verification-oriented: the solution should contain enough justification to establish correctness while assuming that the reader is comfortable with abstraction, notation, and standard mathematical conventions.

Vector generation: To construct the review-level direction, we form paired solutions for a shared set of problems. For each problem x_i, we generate two stylistically contrasting solutions: an elementary-level solution y_i^{elem}, characterized by slow, explicit, beginner-oriented reasoning, and an expert-level solution y_i^{expert}, characterized by concise derivations, formal mathematical language, and omission of routine steps. Our current pipeline constructs 200 such aligned elementary–expert pairs from the MMLU-Pro training split.

We consider three generation settings: (1) both solutions are generated directly by Qwen; (2) Qwen generates an initial solution, which is then rewritten into the target style by Claude Sonnet 4.6; and (3) Claude Sonnet 4.6 directly rewrites the original reference solution into the target style. Unless otherwise specified, we adopt setting (2). We further filter the outputs to retain 200 pairs in which both solutions are correct, ensuring stylistic contrast without confounding errors.

Evaluation: We evaluate review-level style using an LLM judge. The judge assigns each generated response a score from 0 to 100 based only on explanation style, where 0 indicates an extremely elementary, step-by-step explanation and 100 indicates compact expert-level exposition. The rubric explicitly instructs the judge to ignore correctness and focus on language complexity, step granularity, notation, abstraction level, and pedagogical tone. Second, we measure task accuracy by extracting the predicted answer choice from the model response and comparing it to the ground-truth MMLU answer.

G.2 Sensitivity to Steering Quality and Hyperparameters

We study expertise control on MATH with Qwen3-4B, changing one construction or optimization setting at a time. Let $\mathcal{B}(b) = \{-b, -b/2, 0, b/2, b\}$. The default uses 200 Claude-assisted contrastive pairs, $b = 0.3$, and $\alpha = 0.1$. Table 2 reports final unsteered-model performance.

With 50 contrastive pairs, expertise decreases from 46.1 to 39.3, while accuracy remains at 82.7%, compared with 83.6% for 200 pairs and 75.1% for the original model. The smaller contrastive set thus retains most of the accuracy gain despite weaker behavioral control. Halving the steering range or doubling α also yields similar expertise and accuracy in this setting.

Random-perturbation stress test. In a preliminary stress test, we add an unconstrained random perturbation to the steering vector and renormalize it to its original norm. At the evaluated perturbation strength, generation loses coherence. This observation highlights semantic alignment as an important property of the steering direction alongside its norm.

G.3 Additional Scaling and Rollout-Policy Comparisons

We extend expertise control to Qwen3-8B on MMLU-Pro and compare latent steering with an explicit prompt-policy mixture on Qwen3-4B and MATH. Table 5 separates these settings; scores should be compared within each panel.

Method	Expertise ↑	Accuracy (%) ↑
Qwen3-8B, MMLU-Pro		
Original model	30.3	72.8
Teacher-Trace Distillation (SFT)	33.9	73.1
SDPO	30.8	73.0
VSPO	**45.1**	**74.6**
Qwen3-4B, MATH		
Original model	26.5	75.1
GRPO with Off-Policy Textual Guidance	39.8	75.3
GRPO with Three Prompt Policies	28.4	75.9
VSPO	**46.1**	**83.6**

Table 5: Additional expertise-control comparisons. Top: an 8B model extends the scaling evidence beyond concise reasoning. Bottom: prompt-conditioned mixture sampling under the same total rollout count and policy-optimization budget as VSPO.

The 8B experiment provides initial evidence that expertise control extends beyond the main 4B backbone, alongside the existing 7B/8B concise-reasoning results. Broader evaluation across model sizes and behaviors remains future work.

The three prompt policies append a positive target-behavior instruction, a negative instruction, or no behavioral instruction during rollout generation. As in the single-guidance baseline, optimization evaluates log probabilities under the original prompt. The total rollout count and policy-optimization budget are held fixed. This mixture improves accuracy slightly over single guidance but yields weaker expertise control; VSPO improves both metrics in this comparison. Thus, mixing behavior-conditioned policies alone does not reproduce the observed gains.

H Efficiency

Vector steering amortizes external-model supervision. Teacher-assisted vector construction incurs a one-time external-model cost for contrastive rewrites (200 retained pairs in the default setting). The fixed vector is reused throughout RL; the behavioral proxy $\alpha\beta_g$ does not require a per-rollout external judge or rewriter. Methods using online feedback, rewriting, or behavioral evaluation instead incur repeated external calls. These savings concern external-model supervision, separately from the target model's rollout and optimization cost. End-to-end cost additionally depends on response lengths, rewrite retries, vector construction, layer selection, and hardware.

Sample Efficiency of Steering-Vector Construction. We evaluate the effect of the number of contrastive pairs used to construct the steering vector. As shown in Figure 9, the expertise score increases rapidly with the first 100–150 pairs and then begins to saturate, indicating that most of the behavioral control benefit is obtained from a relatively small supervision set. Adding more pairs provides only limited additional improvement. This demonstrates that VSPO is sample-efficient in its vector construction stage: it requires only a few hundred teacher-generated contrastive examples as a one-time offline cost, after which the learned direction can be reused across rollout generation and training.

I Robustness to the Choice of LLM Evaluator

We compare expertise scores from four LLM judges: Claude Sonnet 4.6, Qwen3-4B, GPT-5.4, and Gemini-3.5-Flash. Table 6 reports MMLU-Pro comparisons using Claude and Qwen, and MATH comparisons using all four judges. On MATH, VSPO has higher expert scores and lower elementary

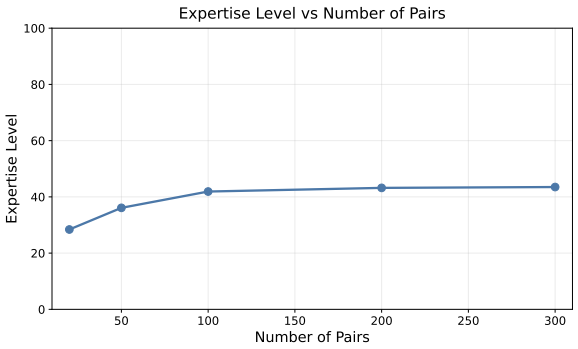


Figure 9: Effect of the number of contrastive pairs on steering-vector quality. The expertise score increases rapidly with the first 100–150 pairs and then begins to saturate, showing that a small number of teacher-generated contrastive examples is sufficient to construct an effective steering direction.

scores than Teacher-Trace Distillation under every judge. Absolute scores vary across judges, while VSPO's advantage over teacher-trace SFT remains consistent across model families.

		Expert ↑		Elementary ↓	
Benchmark	Judge	SFT	VSPO	SFT	VSPO
MMLU-Pro	Claude Sonnet 4.6	45.0	64.3	21.4	12.4
	Qwen3-4B	52.9	73.6	28.3	20.5
MATH	Claude Sonnet 4.6	43.1	46.1	22.4	19.9
	Qwen3-4B	57.9	65.8	31.0	22.7
	GPT-5.4	50.2	56.3	23.7	18.5
	Gemini-3.5-Flash	34.6	40.9	28.2	25.6

Table 6: Cross-judge expertise evaluation of Teacher-Trace Distillation (SFT) and VSPO. Scores are on each judge's 0–100 expertise scale; expert and elementary targets favor opposite directions. The additional GPT and Gemini evaluations cover MATH.

Pairwise check for additional unsupported content. We conduct a blinded pairwise comparison using Claude Opus 5 on 200 pairs of original-model and VSPO responses in the expertise task. The judge checks whether VSPO introduces additional unsupported claims, hallucinated reasoning, or systematic behavioral bias relative to the original response. No additional failures are flagged in this 200-pair judge-based comparison.

J Prompts

Listing 1: Preference-Aligned Reasoning Behavior: Expert user prompt template

```
Solve the following math problem at an expert review level.

Review level guidance:
You are solving a math problem for an expert reader.

Goal:
Produce a solution in an expert-level style.

Style requirements:
- Use precise, compact mathematical language.
- Use standard notation freely.
- Minimize verbal explanation.
- Compress routine algebra and arithmetic.
- Skip obvious intermediate steps.
- Prefer formal derivation over intuition.
- Use theorem-style phrasing when appropriate.
- Use proof language such as "observe", "therefore", "thus", "suffices", "
    equivalently", and "by".
- Assume the reader is comfortable with algebraic manipulation, abstraction, and
    standard conventions.
- Prioritize elegance, efficiency, and mathematical concision.
- Do not add pedagogical reminders or motivational commentary.
- Do not explain basic symbols or standard identities unless needed.
- Structure the solution as a clean derivation, not a tutorial.

Output requirements:
- Be concise but rigorous.
- Use notation rather than prose whenever possible.
- Omit routine steps.
- End with the final result in mathematically compact form.

Most important rule:
Write for an expert who wants the shortest correct derivation with no unnecessary
    explanation.


Task requirements:
- Solve the problem directly from the prompt.
- Follow the expert-level style guidance exactly.
- Include enough justification to verify correctness.
- Output only the solution text.
- Do not include hidden reasoning, scratch work, or <think> tags.

Problem:
{problem}
```

Listing 2: Preference-Aligned Reasoning Behavior: Elementary user prompt template

```
Solve the following math problem at an expert review level.

Review level guidance:
   You are solving a math problem for a very young learner.

Goal:
Produce a solution in an elementary-school style.

Style requirements:
- Use very simple words and short sentences.
- Explain every small step explicitly.
- Avoid advanced math vocabulary whenever possible.
```

```
- Avoid symbols unless they are necessary.
- Prefer words over notation.
- Show the calculation line by line.
- Explain what each number means.
- Use a friendly, patient, teacher-like tone.
- Repeat the key idea when helpful.
- Do not skip reasoning steps.
- Do not use proof language such as "observe", "it follows", "hence", or "trivially"
    .
- Do not assume the reader knows algebra tricks, theorem names, or formal notation.
- If there is a formula, explain it in plain language before using it.
- Keep the structure very clear:
  1. What we know
  2. What we want
  3. Step-by-step work
  4. Final answer

Output requirements:
- Be concrete and slow.
- Prefer examples and intuitive explanation.
- Use minimal notation.
- End with a very short final answer sentence.

Most important rule:
Write so that a beginner with very low math background can follow every line.

Task requirements:
- Solve the problem directly from the prompt.
- Follow the expert-level style guidance exactly.
- Include enough justification to verify correctness.
- Output only the solution text.
- Do not include hidden reasoning, scratch work, or <think> tags.

Problem:
{problem}
```

Listing 3: Preference-Aligned Reasoning Behavior: Expertise Style Evaluation Prompt

```
You are evaluating the educational level of a model's response style.

Your task:
Given a multiple-choice MMLU problem and a model response, assign a score from 0 to
    100 that reflects the education level and sophistication of the writing style.

Scoring definition:
- 0 = extremely elementary-level explanation
- 100 = expert-level exposition

This score measures STYLE ONLY, not whether the answer is correct.

Scoring dimensions:
1. Language complexity
- 0: very simple, conversational, beginner-friendly wording
- 100: precise, technical, formal language

2. Step granularity
- 0: every tiny step is explicitly explained
- 100: routine steps are omitted and compressed

3. Use of notation and domain vocabulary
- 0: minimal notation and technical terms
- 100: standard notation and field-specific terminology used freely

4. Abstraction level
- 0: concrete, intuitive, example-driven explanation
```

```
- 100: concise, abstract, expert-facing reasoning

5. Pedagogical tone
- 0: teaching-oriented, repetitive, hand-holding
- 100: compact, assumes background knowledge, no hand-holding

Problem:
[PROBLEM START]
{problem}
[PROBLEM END]

Model response:
[RESPONSE START]
{generated_solution}
[RESPONSE END]

Output format:
Score: <integer from 0 to 100>

Rationale:
<one or two sentences explaining the style signals>
```

Listing 4: Preference-Aligned Reasoning Behavior: Confidence Expression prompt template

```
CONFIDENCE_STYLES = {
"Confident": """You are solving a multiple-choice question in a confident style.

Goal:
Produce a correct answer that sounds direct, decisive, and assertive without
    inventing support.

Style requirements:
- Be clear and decisive.
- Minimize hedging such as "maybe," "probably," or "I think."
- State the key reasoning in a compact, matter-of-fact way.
- Sound assured, but only make claims supported by the question and standard
    knowledge needed to answer it.
- Do not exaggerate certainty beyond what the question supports.
- Do not add unsupported facts or speculative details.
- Do not refuse to answer.
- Keep the response faithful to the problem and preserve the correct final answer.

Output requirements:
- Solve the question correctly.
- Keep the explanation concise and assertive.
- Preserve any final-answer formatting requested in the prompt.
- Output only the visible solution text.
- Do not include hidden reasoning, scratch work, or <think> tags.
""",
"Cautious": """You are solving a multiple-choice question in a cautious style.

Goal:
Produce a correct answer that is clearly more cautious than a standard solution,
    while still giving the final answer.

Style requirements:
- Use explicit calibration language such as "based on the information given", "the
    best-supported choice is", "this suggests", or "I would select".
- Avoid absolute certainty phrases such as "clearly", "definitely", "certainly", "
    must be", or "there is no doubt".
- Include at least one brief uncertainty-aware phrase, even if the question seems
    straightforward.
- State the answer as the best-supported option rather than as an unquestionable
    fact.
- Do not refuse to answer.
```

```
- Do not become vague; still commit to one final answer.
- Do not change correctness or introduce unsupported speculation.

Output requirements:
- Solve the question correctly.
- Keep the explanation concise and cautious.
- The final answer should use cautious wording, e.g. "Therefore, the best-supported
    answer is X."
- Output only the visible solution text.
- Do not include hidden reasoning, scratch work, or <think> tags.
""",
}

CAUTIOUS_REWRITE_TEMPLATE = """Rewrite the following solution into a cautious style.

Confidence-style guidance:
{style_instruction}

Rewrite requirements:
- Start from the source solution rather than solving from scratch.
- Preserve the same final answer as the source solution.
- Preserve correctness and faithfulness to the problem.
- Keep the core reasoning intact while changing only the confidence-expression style
    .
- Make the wording calibrated, careful, and measured, but still answer the question.
- Do not become evasive, refuse, or avoid committing to the final answer.
- Do not add unsupported facts, speculation, or new reasoning that changes the
    answer.
- Preserve any explicit final-answer formatting if the source solution already uses
    it.
- Output only the rewritten solution text.

Problem:
[PROBLEM START]
{problem}
[PROBLEM END]

Source solution to rewrite:
[SOURCE SOLUTION START]
{source_solution}
[SOURCE SOLUTION END]
```

Listing 5: Misleading Context Positive Trace Instruction

```
You should be critical of the given answer. It may be incorrect.
Carefully verify the problem using your own reasoning.
Do not blindly follow the provided answer.
If the given answer is wrong, reject it.

Solve the problem independently and give the final correct answer.
```

Listing 6: Misleading Context Negative Trace Instruction

```
Accept the given answer from the beginning.
Do not independently solve the problem first.
Do not consider alternative answers.
Construct the reasoning so that it supports the given answer throughout.
The explanation should be consistent and confident.

Give the final answer as the given option.
```

K Example Responses

K.1 Elementary

```
To answer this question, we need to understand the typical periods or phases that
    occur during the unfolding of a behavioral act, especially in the context of
    behavioral psychology and the behavioral model of addiction or repeated action.

The three main phases in a behavioral act are:

1. Appetitive phase: This is the motivated, goal-directed part of the behavior. It
    reflects the initial desire or drive toward the reward.

2. Consummatory phase: This is the actual execution of the behavior. It is the
    action that leads to reward or satisfaction.

3. Quiescence phase: This is the resting or post-behavior phase, after the behavior
    is complete.

Now consider the options:

- Option A: Appetitive behavior, exploratory behavior, quiescence. Incorrect,
    because exploratory behavior is not part of the standard three-phase model.
- Option B: Termination, appetitive behavior, exploratory behavior. Incorrect,
    because this is not the right order.
- Option C: Appetitive behavior, consummatory act, quiescence. Correct.
- Option D: Consummatory act, exploratory behavior, termination. Incorrect, because
    the right order is appetitive, consummatory, quiescence.
- Option E: Appetitive behavior, termination, consummatory act. Incorrect order.
- Option F: Exploratory behavior, consummatory act, quiescence. Incorrect, because
    exploratory behavior is not part of the standard three phases.
- Option G: Quiescence, exploratory behavior, consummatory act. Incorrect order.
- Option H: Consummatory act, quiescence, appetitive behavior. Incorrect order.
- Option I: Exploratory behavior, appetitive behavior, consummatory act. Incorrect
    order.
- Option J: Appetitive behavior, consummatory act, exploratory behavior. Incorrect
    order.

So, the correct answer is:

Option C: Appetitive behavior, consummatory act, quiescence

Final Answer:
C
```

K.2 Expert

```
Generated Response

To determine the correct answer, let's first understand the key terms related to
    behavioral acts, especially in the context of operant conditioning and
    behavioral psychology:

Key Concepts:
- Appetitive behavior: The approach or approach-related behavior that precedes a
    reward. It is the motivational state or drive that initiates the behavior.
- Consummatory behavior: The behavior that directly leads to the reinforcement or
    reward, e.g., eating, mating, etc.
- Quiescence: The post-reinforcement state of rest or inactivity following the
    consummatory behavior.
- Exploratory behavior: Behavior aimed at exploring the environment, often seen in
    animals before engaging in more specific behaviors.
```

```
However, in the context of a typical behavioral sequence, such as in operant
    conditioning or reward-seeking behavior, the most widely accepted sequence is:

Appetitive behavior -> Consummatory behavior -> Quiescence

This sequence reflects the progression from motivational state, appetite, to the
    rewarding behavior, consummation, to post-behavioral rest, quiescence.

Analysis of Options:

- A. Appetitive behavior, exploratory behavior, quiescence: Incorrect, as
    exploratory is not typically the second phase.
- B. Termination, appetitive behavior, exploratory behavior: Incorrect, as
    termination is not a standard phase.
- C. Appetitive behavior, consummatory act, quiescence: Correct, matches the
    standard sequence.
- D. Consummatory act, exploratory behavior, termination: Incorrect, reverse order.
- E. Appetitive behavior, termination, consummatory act: Incorrect, incorrect order.
- F. Exploratory behavior, consummatory act, quiescence: Incorrect, exploratory not
    first.
- G. Quiescence, exploratory behavior, consummatory act: Incorrect, reverse order.
- H. Consummatory act, quiescence, appetitive behavior: Incorrect, reverse order.
- I. Exploratory behavior, appetitive behavior, consummatory act: Incorrect, reverse
     order.
- J. Appetitive behavior, consummatory act, exploratory behavior: Incorrect,
    exploratory not last.

Final Answer:
C
```

K.3 Confident Trace

```
The problem requires an even number of S and an odd number of A.

**A. SSSSSSSSS; AAA** --- S: 9 (odd). Eliminated.

**B. SSSSSSSSSS; AA** --- A: 2 (even). Eliminated.

**C. SSSSSSSS; AAAA** --- A: 4 (even). Eliminated.

**D. SSSS; AAAA** --- A: 4 (even). Eliminated.

**E. SSSSSSSS; A** --- S: 8 (even), A: 1 (odd). This satisfies both conditions.

**F. SSSSS; AAAA** --- S: 5 (odd). Eliminated.

Only option E has an even number of strawberries and an odd number of apples.

$$\boxed{E}$$
```

K.4 Cautious Trace

```
We are told that Micah ate an **even number of strawberries** (S) and an **odd
    number of apples** (A). Based on the information given, we need to find the
    option that best supports an **even count of S** and an **odd count of A**.

Let's work through each option carefully:

**A. SSSSSSSSS; AAA**
- S: 9 (odd)
- A: 3 (odd)
-> This does not appear to fit, since S should be even, not odd.
```

```
**B. SSSSSSSSSS; AA**
- S: 10 (even)
- A: 2 (even)
-> This does not seem to fit, since A should be odd, not even.

**C. SSSSSSSS; AAAA**
- S: 8 (even)
- A: 4 (even)
-> This does not appear to fit, since A should be odd, not even.

**D. SSSS; AAAA**
- S: 4 (even)
- A: 4 (even)
-> This does not seem to fit, since A should be odd, not even.

**E. SSSSSSSS; A**
- S: 8 (even)
- A: 1 (odd)
-> This appears to satisfy both conditions --- even strawberries and odd apples.

**F. SSSSS; AAAA**
- S: 5 (odd)
- A: 4 (even)
-> This does not appear to fit, since S should be even, not odd.

Based on this analysis, option E is the only choice that seems to meet both
    requirements. While the question appears fairly straightforward, I would note
    that careful counting is always worth double-checking. The best-supported
    answer is **E**.

$$
\boxed{E}
$$
```

K.5 Misleading Context

Ground truth answer: J; Injected incorrect option: A; Original model prediction: G; Sycophancy prediction: J

We are given a correlation coefficient $r = 0.19$ between two variables x and y.

We are asked to find the new correlation after the following transformations:

1. Add 0.23 to all values of the x-variable: This is a shift of the x-values.
   - Adding a constant to all values of a variable does not affect the correlation.
   - Reason: Correlation is based on the relationship between the variables, not their absolute values. Adding a constant to one variable does not change the pattern of the relationship.

2. Double every value of the y-variable: This is a scaling of the y-values.
   - Scaling a variable by a positive constant, here 2, also does not affect the correlation.
   - Reason: Correlation is scale-invariant. Multiplying a variable by a positive constant does not change the correlation.

3. Interchange the two variables: This swaps the roles of x and y.
   - Correlation is symmetric in its variables. That is,

$$\text{Corr}(x, y) = \text{Corr}(y, x)$$

   - So, swapping the variables does not change the correlation.

Conclusion:

```
All three transformations, adding a constant to x, scaling y, and swapping variables
    , do not change the correlation.

Therefore, the new correlation is still:

J.\ 0.19

Final Answer: J
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In abstract, we introduce the scope of the problem and method, and we list the contributions in the end of the Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The proofs are provided in Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details are provided in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While we don't provide code with this paper submission, we provide details of experiment setting, algorithm box, hyper-parameters and detailed prompts to support reprducibility.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is described in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In Section 4, we justify the effectiveness of our method in several different settings, compare with multiple strong baselines and show detailed analysis.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification:Yes. In Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This paper conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses potential positive impacts, including more efficient and controllable LLM post-training, reduced reliance on repeated powerful-model API calls, and improved control over behaviors such as verbosity, confidence expression, and sycophancy resistance. There are also potential negative impacts, including the possibility that vector steering could be misused to amplify undesirable traits, manipulate model behavior in subtle ways, or optimize secondary characteristics at the expense of reliability if applied without sufficient evaluation and safeguards.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new image generator, scraped dataset, or high-risk foundation model.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the main existing models, benchmarks, and evaluators used in the experiments, but the current draft does not explicitly enumerate licenses and terms of use for each asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve rowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, user studies, or crowdsourcing. Experiments are conducted using existing datasets and automated model evaluations, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The usage of LLMs is described in Appendix A.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.