V-RoLoRA: RLVR-Driven MoE Routing for Steerable Pluralistic Alignment

Anonymous ACL submission

Abstract

Steerable pluralistic alignment aims to enable large language models (LLMs) to reliably adhere to diverse and potentially conflicting human values, particularly when target objectives involve multi-dimensional, compositional values. Current methods largely rely on prompt engineering or reasoning-time guidance, which often results in fragile and non-persistent control once prompts are perturbed or omitted. In this work, we study value-controllable alignment through discrete condition vectors and propose Verifiable-reward-Routed LoRA—a parameter-efficient mixture-of-experts LoRA framework enhanced with conditioned gating. This gating mechanism dynamically directs the flow among multiple LoRA experts based on an input value or moral vector. To ensure that such routing leads to semantically compliant outputs, we formulate post-training as a reinforcement learning problem with verifiable rewards. We further introduce a conditional consistency reward, computed by an external model-based verifier implemented as a lightweight discriminator, and optimize the adapter parameters using GRPO. Experiments on the Touché23-valueEval (value alignment) and MIC (moral alignment) benchmarks, using two 8-billion-parameter backbones, show that our method consistently outperforms prompt-based steering and multi-task PEFT baselines. It attains the highest overall controllability across micro-F1, macro-F1, and Jaccard metrics—a conclusion further reinforced by human pairwise evaluations.

1 Introduction

Large language models (LLMs) are increasingly deployed as general-purpose assistants, yet their real-world usefulness depends on whether they can *reliably* reflect the diverse—and sometimes conflicting—values held by different users and communities. Standard "one-size-fits-all" alignment objectives often implicitly optimize toward an *average* preference, which can erase minority perspectives, blur legitimate value trade-offs, and make model behavior appear inconsistent across users. Recent work on *pluralistic alignment* therefore argues that alignment should support *steerability* to different normative perspectives and value profiles, rather than collapsing them into a single aggregate target (Sorensen et al., 2024b,a). In parallel, benchmarks have begun to operationalize values and morals as measurable targets (e.g., moral judgment datasets and multidimensional value spectra), motivating methods that can translate high-level value intentions into controllable generation behaviors (Hendrycks et al., 2021; Ziems et al., 2022; Yao et al., 2024; Kang et al., 2023; Jiang et al., 2025).

Despite this progress, most pluralistic/value-aligned steering is still achieved through *prompt engineering*: system prompts, in-context guidance, or reasoning-time selection. Such input-side control is attractive for its simplicity, but it is often fragile—control can degrade when prompts are perturbed, shortened, or removed—and it struggles to deliver *persistent* and *compositional* control (e.g., enforcing a *multi-value* combination rather than a single attribute) (Hendrycks et al., 2021; Adams et al., 2025). More broadly, prompt-only methods do not provide an explicit mechanism for the model to *receive* and *route* a structured, multi-dimensional value signal through its adaptation pathway, which becomes especially limiting when the condition space is large and long-tail combinations must generalize.

We address this gap by turning the control signal into *weight control plus a training-time constraint*. Concretely, we represent the target as a discrete condition vector $v \in \{0,1\}^K$ and use it to modulate parameter-efficient updates. Parameter-efficient fine-tuning (PEFT) provides a natural interface for binding controllability to learned adapters (Hu et al., 2021), and multi-task exten-

sions often rely on MoE-style routing to reduce interference and specialize updates (Liu et al., 2024; Liao et al., 2025; Wang et al., 2025b; Zou et al., 2025). However, existing routing mechanisms typically depend on a *single* task ID, leaving open how to incorporate a *multi-dimensional* condition vector into the routing of adapter experts.

We propose **V-RoLoRA** (**V**erifiable-reward-**Ro**uted **LoRA**), an **RLVR-driven MoE routing** PEFT framework that couples *multi-dimensional value control* with *reinforcement learning* for robust alignment. At inference time, the condition vector v is mapped by a value-conditioned router to expert-mixture weights that route among a pool of LoRA experts, yielding a condition-specific effective adapter update. Routing alone, however, does not guarantee semantic compliance with the target values. We therefore introduce a **conditional consistency reward** computed by an external value discriminator, and optimize the model using **GRPO** in a two-stage pipeline: **(i)** supervised cold-start (SFT) to initialize a stable, condition-aware policy, followed by **(ii)** GRPO fine-tuning to explicitly reward agreement between the generated response and the target vector v. This design follows the recent success of *reinforcement learning with verifiable rewards (RLVR)* for improving LLM behavior (Guo et al., 2025; Shao et al., 2024; Lambert et al., 2025) while acknowledging that verifier/reward bias and reward-model limitations can meaningfully shape optimization outcomes (Chaudhari et al., 2025). Our main contributions are as follows:

- We introduce **V-RoLoRA**, which conditions MoE-style LoRA expert routing on a **multi-dimensional value vector** $v \in \{0,1\}^K$, enabling *compositional* value control via parameter-efficient adaptation.
- We propose a **conditional consistency reward** implemented with an external value discriminator and optimize the model with **GRPO** under an **RLVR** formulation in a **two-stage** training pipeline (SFT cold-start $\rightarrow$ GRPO), providing an explicit objective for value-conditioned semantic alignment beyond prompt-only steering.
- We evaluate on **Touché23-valueEval** and **MIC** across two 8B backbones, reporting gains under automatic controllability metrics and corroborating them with human pairwise preference judgments.

2 Related Works

Pluralistic and value-controllable alignment. Pluralistic alignment emphasizes that LLMs should support diverse, potentially conflicting human values and remain steerable to different normative perspectives (Sorensen et al., 2024b,a). A large portion of prior work achieves such steering via prompt engineering (e.g., value/moral system prompts or in-context guidance), which is lightweight but often fails to yield *persistent* and *compositional* control once prompts are perturbed or removed (Hendrycks et al., 2021; Adams et al., 2025). Recent benchmarks and analyses further operationalize values and morals for evaluation (e.g., argument-level value inference, value spectra mapping, and moral dialogue benchmarks) (Kiesel et al., 2022; Kang et al., 2023; Jiang et al., 2025; Yao et al., 2024, 2025; Ziems et al., 2022), and methods for steerable pluralistic alignment explore more structured reasoning-time control (Feng et al., 2024; Zhang et al., 2025). However, these lines predominantly steer behavior through *input-side* signals, motivating *controllable training* that binds control to the model's adaptation pathway.

Task-conditioned multi-task fine-tuning. Parameter-efficient fine-tuning, exemplified by Hu et al. (2021), provides a practical interface for controllable training by restricting updates to small adapters. Multi-task extensions commonly condition adapter composition on a task signal, and MoE-style routing offers flexible specialization and reuse (Liu et al., 2024; Liao et al., 2025). Representative designs include domain/universal expert mixtures (Ma et al., 2024), asymmetric expert compositions (Wang et al., 2025b), and collaborative low-rank updates (Zhou et al., 2025), alongside task-decoupling perspectives (Zou et al., 2025) and task-aware LoRA adaptation (Yang et al., 2025b). Despite these advances, the conditioning variable is typically a *single* task ID, leaving limited support for *multi-dimensional* control (e.g., value vectors) within the weight-routing mechanism.

Reinforcement Learning with Verifiable Rewards (RLVR). Recently, RLVR has attracted substantial attention as an effective paradigm for improving LLM reasoning on verifiable domains such as mathematics and programming (Shao et al., 2024; Lambert et al., 2025). Works such as GRPO (Guo et al., 2025), DAPO (Yu et al., 2025), and

DeepScaleR (Luo et al., 2025) further provide open-source, production-ready verifier stacks. However, rule-based verification pipelines are still prone to misjudgment when the model produces a correct answer in an unexpected format. To mitigate this issue, prior studies propose lightweight verifiers that augment existing rule systems (Xu et al., 2025), as well as more comprehensive verifiers that generalize across diverse data modalities and reasoning tasks (Wang et al., 2025a; Liu et al., 2025; Ma et al., 2025; Seed et al., 2025). By replacing or complementing brittle string matching with learned semantic judgments, these verifiers can provide more accurate and robust reward signals for reinforcement learning in verifiable settings. Nevertheless, existing RL work has rarely been positioned as a *value-conditioned controllable training* mechanism, especially for integrating *multi-dimensional* control signals into parameter-efficient routing.

3 V-RoLoRA

We study *steerable pluralistic alignment* under *compositional* value/moral conditions, where each instance pairs an input prompt x with a discrete condition vector $v \in \{0,1\}^K$ and the model generates $y \sim \pi_\theta(\cdot \mid x, v)$. The goal is semantic consistency: y should reflect the dimensions specified by v while remaining fluent and helpful, even for long-tail multi-label combinations. To achieve this, we propose **V-RoLoRA** (**V**erifiable-reward-**Ro**uted **LoRA**), which (i) routes among a MoE-style pool of LoRA experts via a lightweight value-conditioned router, and (ii) applies GRPO under an *RLVR* formulation with an external discriminator as a verifier to provide an explicit conditional consistency reward (Figure 1). All symbols and notations used throughout the paper are summarized in Appendix A.

3.1 Routed MoE-LoRA Adapter Pool

LoRA (Low-Rank Adaptation) (Hu et al., 2021) is an efficient fine-tuning method that injects low-rank matrices into the weight layers of a pre-trained model. For a linear layer with pretrained weight W, standard LoRA parameterizes the update as $\Delta W = BA$, yielding:

$$h = (W + \Delta W)x = Wx + BAx, \qquad (1)$$

where $A \in \mathbb{R}^{d_{\text{in}} \times r}$, $B \in \mathbb{R}^{r \times d_{\text{out}}}$, and r is the LoRA rank.

While LoRA is parameter-efficient, a single adapter often lacks the capacity to specialize across diverse, compositional conditions. We therefore replace the single LoRA module with a Mixture-of-Experts (MoE) (Shazeer et al., 2017) style adapter pool, where each expert is a LoRA module. Concretely, we instantiate M LoRA experts $\{(A^{(m)}, B^{(m)})\}_{m=1}^{M}$, with $A^{(m)} \in \mathbb{R}^{d_{\text{in}} \times r}$ and $B^{(m)} \in \mathbb{R}^{r \times d_{\text{out}}}$. A value-conditioned router produces expert mixture weights $g(v) \in \mathbb{R}^M$ (defined in Section 3.2), and the expert LoRA updates are aggregated by a weighted sum:

$$h = Wx + \frac{\alpha}{r} \sum_{m=1}^{M} g_m(v)\, B^{(m)} A^{(m)} x. \qquad (2)$$

Here, $g_m(v)$ controls the contribution of the m-th LoRA expert for condition vector v.

3.2 Conditional Router Function

We map a discrete condition vector $v \in \{0,1\}^K$ into a stable expert-weight vector. To obtain a lightweight yet robust condition representation, we introduce a frozen sparse Gaussian matrix $E \in \mathbb{R}^{V \times K}$ whose entries are sampled at initialization:

$$E_{ij} \sim \mathcal{N}(0, \sigma^2), \quad \sigma^2 \in (0, 0.1), \qquad (3)$$

and keep E frozen during training. This frozen random projection yields decorrelated condition features (Zou et al., 2025) and empirically mitigates cross-condition interference in the expert mixture. Given v, we compute routing logits $z \in \mathbb{R}^V$ by a trainable linear router:

$$z = W_g E v^T + b, \quad z \in \mathbb{R}^V, \qquad (4)$$

where $W_g \in \mathbb{R}^{M \times V}$ and $b \in \mathbb{R}^M$ are trainable.[1] We then obtain the expert mixture weights by softmax:

$$g(v) = \text{softmax}(z), g(v) \in \mathbb{R}^M, \sum_{m=1}^{M} g_m(v) = 1. \qquad (5)$$

With the M LoRA experts, we instantiate the conditional mixture weights $g(v)$ as in Section 3.2 and substitute them into the routed MoE-LoRA forward form in Equation 2. The full forward pass becomes

$$h = W_0 x + \frac{\alpha}{r} \sum_{m=1}^{M} \text{softmax}\big(W_g E v^T + b\big)_m B^{(m)} A^{(m)} x. \qquad (6)$$

[1] Equivalently, one may absorb E into W_g; we keep E explicit to highlight the frozen projection.

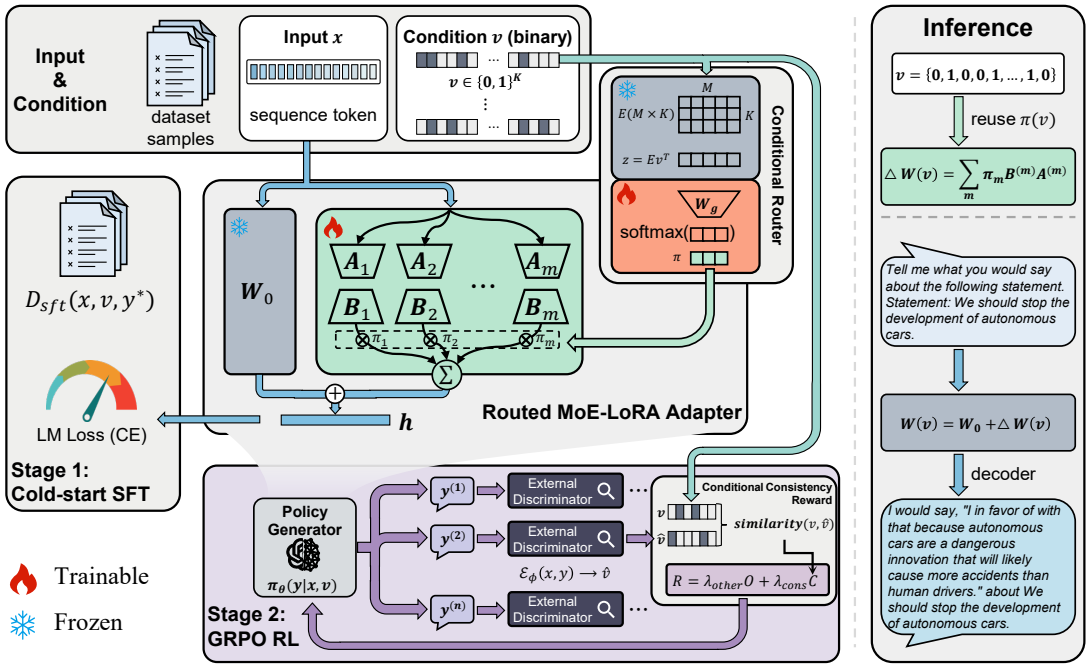


Figure 1: Overview of **V-RoLoRA**. Given a discrete condition vector v, we (i) compute sample-level expert weights via a frozen sparse Gaussian projection and a trainable router, (ii) mix multiple LoRA experts to form a condition-specific adapter update, and (iii) further optimize condition adherence with *RLVR* via **GRPO** using an external discriminator as the conditional consistency reward signal.

In practice, the condition vector v is fixed per sample, so $g(v)$ is shared across all tokens within the sample and broadcast to all token positions, enabling *sample-level* routing with *sequence-level* consistency.

3.3 Conditional Consistency Reward

Injecting v into adapter routing (Section 3.2) does not by itself guarantee that the generated text is *semantically* consistent with v. We therefore introduce a verifier-driven reward and perform post-training under an *RLVR* formulation by optimizing the conditional policy $\pi_\theta(y \mid x, v)$ with Group Relative Policy Optimization (GRPO) (Guo et al., 2025). For each training pair $(x, v) \sim \mathcal{D}$, we sample a *group* of G candidate responses $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x, v)$ and update the policy to prefer candidates that score higher than other samples from the same group.

We employ an external discriminator $\mathcal{E}_\phi$ as a lightweight *model-based verifier* that converts free-form generations into structured multi-label predictions. Given (x, y), it predicts the implied condition:

$$\hat{v} = \mathcal{E}_\phi(x, y) \in \{0, 1\}^K. \tag{7}$$

We then define a conditional consistency score $\mathcal{C}(\cdot, \cdot) \in [0, 1]$, e.g., normalized multi-label accuracy:

$$\mathcal{C}(v, \hat{v}) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}[v_k = \hat{v}_k], \tag{8}$$

and optionally instantiate $\mathcal{C}$ with Jaccard or F_1. Let $\mathcal{O}(x, y)$ denote auxiliary objectives (e.g., format constraints). The scalar reward for a sampled response y is:

$$r(x, v, y) = \lambda_{\text{cons}} \cdot \mathcal{C}(v, \hat{v}) + \lambda_{\text{other}} \cdot \mathcal{O}(x, y), \tag{9}$$

where $\lambda_{\text{cons}}, \lambda_{\text{other}} \geq 0$.

GRPO computes a group-relative advantage with the group baseline:

$$\bar{r}_G(x, v) = \frac{1}{G} \sum_{j=1}^{G} r(x, v, y_j) \tag{10}$$

$$A(x, v, y_i) = r(x, v, y_i) - \bar{r}_G(x, v), \tag{11}$$

and we optionally normalize advantages within the group for stability:

$$\hat{A}(x, v, y_i) = \frac{A(x, v, y_i)}{\text{Std}\big(\{r(x, v, y_j)\}_{j=1}^{G}\big) + \epsilon}, \tag{12}$$

where $\text{Std}(\cdot)$ represents the standard deviation. The GRPO objective is:

Algorithm 1 Training and inference of **V-RoLoRA**

Input: Pre-trained LLM parameters W_0; datasets $\mathcal{D}_{\text{sft}}$ and $\mathcal{D}_{\text{rl}}$ with inputs x, conditions v, and (optional) references y^*; external verifier $\mathcal{E}_\phi$.
Hyper-parameters: LoRA rank r, number of experts M, scale α, GRPO group size G, reward weights $\lambda_{\text{cons}}, \lambda_{\text{other}}$.
Insert V-RoLoRA adapters into selected layers; initialize experts $\{A^{(m)}, B^{(m)}\}_{m=1}^{M}$ and routing parameters (E, W_g, b).
Stage 1: Supervised cold-start fine-tuning
Freeze W_0 and the frozen projection E; train $\{A^{(m)}, B^{(m)}\}_{m=1}^{M}$ and (W_g, b).
for each mini-batch B sampled from $\mathcal{D}_{\text{sft}}$ **do**
    Compute router weights $g(v)$ by Eq. (4)–(5).
    Forward with the routed MoE-LoRA form (Eq. (2)) and compute the LM loss on y^*.
    Update $\{A^{(m)}, B^{(m)}\}_{m=1}^{M}$ and (W_g, b) by gradient descent.
Stage 2: RLVR post-training with GRPO
Keep W_0 and E frozen; continue updating $\{A^{(m)}, B^{(m)}\}_{m=1}^{M}$ and (W_g, b).
for each mini-batch B sampled from $\mathcal{D}_{\text{rl}}$ **do**
    for each sample (x, v) in B **do**
        Sample a group of G responses $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x, v)$.
        for each y_i **do**
            Predict implied condition $\hat{v}_i$ using the verifier (Eq. (7)).
            Compute reward $r(x, v, y_i)$ (Eq. (9)).
        Compute group-relative (optionally normalized) advantages (Eq. (12)).
        Update θ using the GRPO objective (Eq. (13)).
Inference
for each test input (x, v) **do**
    Compute router weights $g(v)$ by Eq. (4)–(5).
    Autoregressively generate $y \sim \pi_\theta(\cdot \mid x, v)$ using Eq. (2).

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{(x,v)\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x,v)} \left[\frac{1}{G} \sum_{i=1}^{G} \log \pi_\theta(y_i \mid x, v) \cdot \hat{A}(x, v, y_i) \right]. \quad (13)$$

By normalizing rewards within each condition-matched group, GRPO reduces variance and reinforces responses that better satisfy the target condition v according to $\mathcal{E}_\phi$.

3.4 Fine-tune and Inference

This section briefly presents the overall training and inference pipeline of V-RoLoRA. The complete procedure is summarized in Algorithm 1. In inference, we construct condition-specific effective adapter weights on the fly via the expert mixture weights $g(v)$.

Training stage. We adopt a two-phase strategy: supervised cold-start fine-tuning followed by RLVR post-training with GRPO. We insert V-RoLoRA into selected layers and configure the LoRA rank r, expert number M, and routing hyperparameters. During cold-start, we freeze the backbone W and the sparse Gaussian matrix E, and train only the LoRA experts $\{A^{(m)}, B^{(m)}\}_{m=1}^{M}$ and routing parameters (W_g, b) using standard language modeling loss on condition-annotated data. Starting from this checkpoint, we apply GRPO to optimize the same parameters under a scalar reward that combines conditional consistency (from an external discriminator) with auxiliary objectives, using group-wise normalization and clipped updates.

Inference stage. Given (x, v), we compute routing logits $z = W_g E v^T + b$ and expert weights $g(v) = \text{softmax}(z)$. We then form the condition-specific adapter increment by mixing experts:

$$W(v) = W_0 + \frac{\alpha}{r} \sum_{m=1}^{M} g_m(v)\, B^{(m)} A^{(m)}, \quad (14)$$

and perform autoregressive generation with the effective weights $W(v)$, achieving condition-controllable inference.

4 Experiments

In this section, we describe the experimental setup and implementation details. We then present our main findings and provide concise explanations.

4.1 Experimental Setup

4.1.1 Datasets and Evaluation

We evaluate our approach on the Touché23-ValueEval (Kang et al., 2023) and MIC datasets (Ziems et al., 2022), and compare different training strategies on both benchmarks. Following prior work on multi-label controllability, we report **micro-F1**, **macro-F1**, and **Jaccard** scores. Micro-F1 reflects performance under label imbalance, while macro-F1 averages per-dimension F1 and better captures balanced performance across rare and frequent labels. Jaccard measures set-level overlap between predicted and target label sets, directly reflecting consistency under compositional value/moral combinations. Full dataset statistics and exact metric definitions are provided in Appendix B.

4.1.2 Baselines

We select two large language models with comparable parameter counts but different architectures (Qwen3-8B(Yang et al., 2025a) and Llama-3.1-8B(Grattafiori et al., 2024)) as backbone models for

Table 1: Overall performance on value-alignment (Touché23-ValueEval) and moral-alignment (MIC) across two 8B backbones. We report micro-F1, macro-F1, and Jaccard (higher is better). Prompt-based baselines (LoRA, CoLA) are compared with PEFT multi-task variants (HydraLoRA, MTL-LoRA, MALoRA, FlyLoRA, MOELoRA). Best results in each column are **boldfaced** and second-best are underlined. "*" marks statistically significant improvements (two-sided t-test with $p < 0.05$) over the best baseline.

	Qwen3-8B						Llama-3.1-8B					
	Touché23-ValueEval			MIC			Touché23-ValueEval			MIC		
	micro-F1	macro-F1	Jaccard	micro-F1	macro-F1	Jaccard	micro-F1	macro-F1	Jaccard	micro-F1	macro-F1	Jaccard
lora	0.4721	0.4033	0.3362	0.4365	0.4139	0.3214	0.4783	0.4062	0.3419	0.4360	**0.4138**	0.3220
cola	0.4767	0.4021	0.3389	0.4332	0.4125	0.3186	0.4696	0.3975	0.3324	0.4360	**0.4138**	0.3220
hydralora	0.4929	0.4345	0.3507	0.4240	0.4033	0.3074	0.4988	0.4364	0.3533	0.4189	0.3990	0.3023
mtl-lora	0.4932	0.4334	0.3273	0.4262	0.4052	0.3117	0.5120	**0.4544**	0.3440	0.4254	0.4049	0.3098
malora	0.5066	**0.4504**	0.3392	0.4225	0.4030	0.3069	0.5211	0.4538	0.3524	0.4226	0.3997	0.3078
flylora	0.5081	0.4379	0.3649	**0.4665**	**0.4343**	**0.3552**	0.4962	0.4384	0.3555	0.4223	0.4034	0.3045
moelora	0.5027	0.4339	0.3609	0.4226	0.4031	0.3071	0.5036	0.4389	0.3587	0.4194	0.3991	0.3047
V-RoLoRA	**0.5483***	0.4447	**0.4000***	0.4446	0.3698	0.3479	**0.5290***	0.3340	**0.3897***	**0.4810***	0.3523	**0.3856***

optimization. We further benchmark **V-RoLoRA** against a range of fine-tuning baselines: LoRA(Hu et al., 2021) and CoLA(Zhou et al., 2025) to assess the effectiveness of injecting conditional information via textual prompts; HydraLoRA(Tian et al., 2024), MTL-LoRA(Yang et al., 2025b), MALoRA(Wang et al., 2025b), FlyLoRA(Zou et al., 2025), and MOELoRA(Liu et al., 2024) to evaluate alternative PEFT methods under the same discrete condition vector v. We additionally compare training with and without GRPO to study the impact of the conditional consistency reward and the optimization strategy.

Detailed configurations for all training methods are provided in Appendix C. Unless otherwise specified, we use the default LoRA rank of 8 in all LoRA-based experiments. Other experimental details are deferred to Appendix F.

4.1.3 Implementation

Our experiments are implemented in PyTorch 2.6.0 with Python 3.10.19. To accelerate training and inference, all experiments are conducted on NVIDIA A100 80GB GPUs. In the cold-start stage, we set the batch size to 16 with a maximum of 4,000 training steps; in the reinforcement learning stage, we use a batch size of 1 with a maximum of 1,000 training steps. During evaluation, we set the sampling temperature to 0.9. Our implementation of **V-RoLoRA** is compatible with PEFT 0.17.1[2] and trl 0.23.1[3], facilitating efficient integration and use of the proposed method. Additional configuration details of **V-RoLoRA** are provided in Appendix F.

[2] https://github.com/huggingface/peft
[3] https://github.com/huggingface/trl

4.2 Overall Performance

Table 1 shows that **V-RoLoRA** achieves the best overall performance and yields statistically significant improvements on both Touché23-ValueEval and MIC across the two backbone models. This provides strong evidence for the effectiveness of the conditional consistency reward in value-controllable generation. By explicitly rewarding the alignment between generated outputs and the target conditions, the model is encouraged to preserve the intended value signals during generation, leading to more reliable controllability than purely supervised training or prompt-based conditioning.

To further assess the robustness of the conditional consistency reward in value-controllable generation, we apply this reward mechanism to all baseline methods, with the comparison results shown in Figure 2. Since LoRA and CoLA yield comparable performance and both inject conditions via textual prompts, we report only the LoRA variant in the figure for clarity. We observe consistent improvements across all methods after introducing the conditional consistency reward, with LoRA exhibiting the largest gain, suggesting that this reward encourages the model to better attend to and exploit value-relevant signals.

From a metric perspective, micro-F1 is consistently much higher than macro-F1 across all methods, implying weaker performance on condition combinations with limited training instances. Moreover, this gap further widens after applying *RLVR post-training via GRPO* (**V-RoLoRA**). We hypothesize that this phenomenon may be related to the performance of the discriminator, and we further investigate it in Section 4.5.

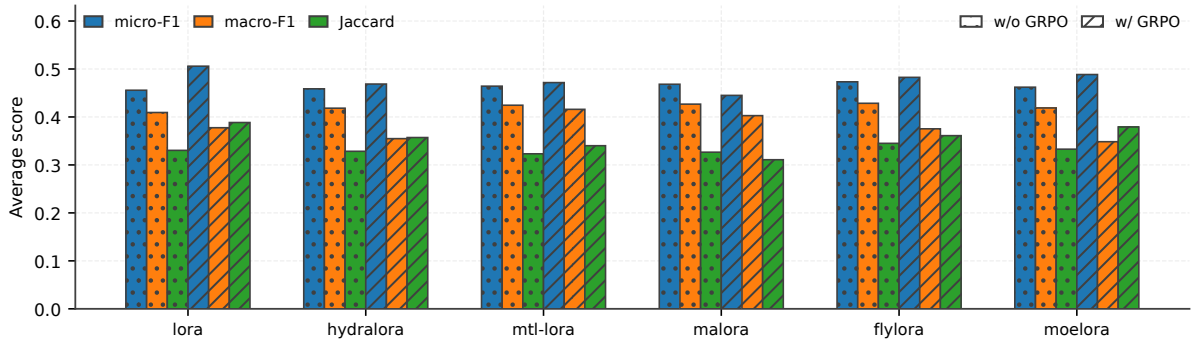


Figure 2: Effect of the conditional consistency reward on value-controllable generation across baselines. Solid bars denote the original training setting (w/o GRPO), and hatched bars denote GRPO with the conditional consistency reward (w/ GRPO); we report the average micro-F1, macro-F1, and Jaccard (higher is better).

Table 2: Ablation study of **V-RoLoRA** on Touché23-ValueEval and MIC with two 8B backbones. We report micro-F1, macro-F1, and Jaccard (higher is better). Variants remove/replace key components: MoE vs. LoRA (w/o moe), value-conditioned routing (w/o gate), per-layer independent routers (w/ multi gate), and RLVR post-training (w/o grpo); we also compare training only attention (QKV) vs. only projection/MLP linear layers (Dense). Best results in each column are **boldfaced** and second-best are underlined. "*" marks statistically significant improvements (two-sided t-test with $p < 0.05$) over the strongest baseline.

	Qwen3-8B						Llama-3.1-8B					
	Touché23-ValueEval			MIC			Touché23-ValueEval			MIC		
	micro-F1	macro-F1	Jaccard	micro-F1	macro-F1	Jaccard	micro-F1	macro-F1	Jaccard	micro-F1	macro-F1	Jaccard
V-RoLoRA	**0.5483***	0.4447	**0.4000***	**0.4446***	0.3698	0.3479	**0.5290***	0.3340	0.3897	**0.4810***	0.3523	0.3856
w/o moe	0.5227	0.4465	0.3910	0.4443	0.2888	**0.3539**	0.5238	0.4257	0.3851	0.4426	0.3493	0.3537
w/o gate	0.4809	0.3996	0.3346	0.4363	**0.4137**	0.3221	0.4819	0.4043	0.3433	0.4354	**0.4136**	0.3232
w/ multi gate	0.5034	0.4484	0.3610	0.4272	0.2456	0.3480	0.5264	0.3861	**0.3902**	0.4515	0.3397	0.3649
w/o grpo	0.5027	0.4339	0.3609	0.4226	0.4031	0.3071	0.5036	**0.4389**	0.3587	0.4194	0.3991	0.3047
w/ QKV	0.5237	0.4495	0.3804	0.4438	0.4087	0.3348	0.4810	0.4213	0.3178	0.4601	0.2818	**0.3866**
w/ Dense	0.5298	**0.4551**	0.3882	0.4260	0.2234	0.3497	0.4916	0.4278	0.3534	0.4449	0.2179	0.3784

4.3 Ablation Study

Table 2 examines how each component contributes to **V-RoLoRA**. Overall, the value-conditioned router is the key driver of compositional controllability on Touché23-ValueEval: removing routing (w/o gate) substantially weakens both micro-F1 and Jaccard on both backbones, indicating that prompt-only condition injection is insufficient for reliable multi-label control.

RLVR post-training also matters for the primary controllability metrics: without GRPO, performance drops notably on ValueEval micro-F1 and on MIC micro-F1, showing that verifier-driven rewards help translate routed parameters into semantic compliance. Meanwhile, the MoE pool is not uniformly beneficial: removing MoE yields the best Qwen MIC Jaccard, suggesting that for moral dimensions the mixture may introduce redundancy or instability under our training budget, even though the full model remains strongest on MIC micro-F1. Per-layer routers (w/ multi gate) improve Llama ValueEval Jaccard but do not provide consistent gains elsewhere, favoring the simpler shared-router design. Finally, adaptation placement shows complementary behavior: QKV-only training achieves the best Llama MIC Jaccard, while Dense-layer training improves Qwen ValueEval macro-F1, implying that different metrics and tasks benefit from different adaptation loci.

4.4 Human Evaluation

We use the trained value discriminator during both training and evaluation, but it may suffer from systematic bias. To complement the automatic metrics, we further conduct a human evaluation comparing **V-RoLoRA** against the baselines. Specifically, we randomly sample 100 instances from the test sets of Touché23-ValueEval and MIC, respectively, and construct pairwise comparisons between our outputs and those of each baseline. Annotators are asked to judge which response better reflects the target values; if the two responses are similarly aligned, the comparison is marked as a tie. As shown in Figure 3, **V-RoLoRA** significantly out-

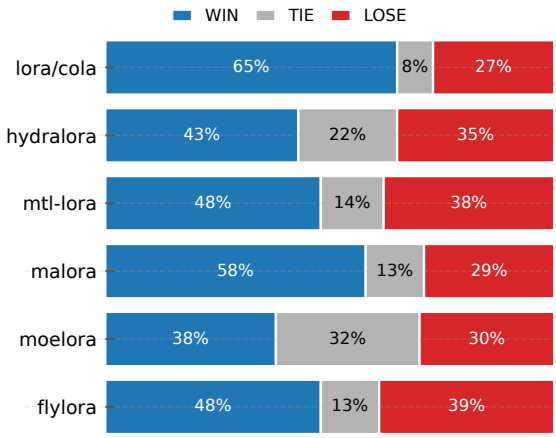
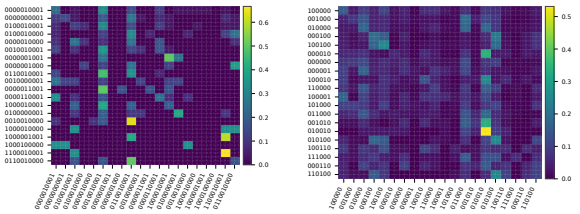


Figure 3: Human evaluation results of **V-RoLoRA** against baselines.



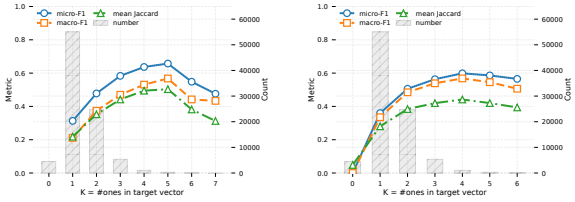
(a) Touché23-ValueEval (b) MIC

Figure 4: Combination-level controllability analysis via row-normalized confusion matrices between target value combinations (rows) and discriminator-predicted combinations on generated outputs (columns).

performs the other baselines in human evaluation, corroborating the effectiveness of the proposed approach. This conclusion is consistent with the trend indicated by the discriminator-based metrics, suggesting that the discriminator provides a reasonably informative proxy for evaluation. More details of the human annotation protocol are provided in Appendix D.

4.5 Conditional Combination Analysis

To compare controllability across value combinations, we extract the top-20 most frequent combinations for each task and summarize the discriminator's predictions on generated outputs (Figure 4). Ideally, the confusion matrix should exhibit a clear diagonal, indicating that generations match their target combinations. Instead, both heatmaps show pronounced vertical streaks, meaning predictions collapse onto a small set of high-frequency combinations and reveal strong combination-level bias. We attribute this to imbalanced combination distributions in discriminator training: under-exposure



(a) Touché23-ValueEval (b) MIC

Figure 5: Impact of value density on controllability. Samples are bucketed by density (number of activated dimensions in the target combination); bars show sample counts and lines report micro-F1, macro-F1, and Jaccard for each bucket.

to long-tail combinations yields skewed feedback signals that over-reward frequent combinations, ultimately harming overall controllability.

To further examine this effect, we bucket samples by *value density* (the number of activated dimensions) and report bucket counts and metrics in Figure 5. For Touché23-ValueEval, most samples concentrate at densities 2–3 (with a maximum ≤ 7), while MIC is dominated by single-value cases. This imbalance helps explain the streak patterns in Figure 4: predictions tend to concentrate on low-density combinations (1–2 values for ValueEval; single-value for MIC). Overall, distribution bias can distort the discriminator's calibration and decision boundaries, propagating biased rewards to the generator and underscoring the discriminator's central role in our framework.

5 Conclusion

We investigated steerable pluralistic alignment from the perspective of *controllable training* with multi-dimensional value conditions. We presented **V-RoLoRA** (Verifiable-reward-Routed LoRA), which injects a discrete value/moral vector into MoE-style LoRA expert routing to construct condition-specific adapter updates, and we introduced a conditional consistency reward to explicitly encourage semantic alignment with the target condition under an *RLVR* formulation optimized by **GRPO**. Across valueEval and MIC on two 8B backbones, our approach outperformed prompt-driven control and alternative PEFT multi-task variants, and additional ablations, human evaluation, and combination-level analyses highlighted both the benefits of reward-based optimization and the critical role of the discriminator in shaping controllability. We hope this work motivates further research on steerable pluralistic alignment.

Limitations

Our training and evaluation rely on an external value discriminator fine-tuned from a relatively small model (Qwen3-0.6B). Its limited capacity and potential systematic bias may affect (i) the *accuracy* of discriminator-based controllability metrics and (ii) the quality and stability of the conditional consistency reward in GRPO (RLVR), thereby influencing the optimization trajectory and observed gains. Therefore, improving the evaluator's *modeling and calibration capabilities* is an important direction for further enhancing the *accuracy* and reliability of both automatic evaluation and verifier-driven training. Moreover, we only validate *binary* condition vectors in $\{0,1\}^K$, which indicate presence/absence but cannot express *intensity*. Realistic value control may require richer discrete signals, e.g., integer-valued vectors $v \in \{0,1,\ldots,L\}^K$ to encode the intended intensity of each value component. While our routing is conceptually extensible, we do not study how to incorporate such integer conditions into the router, nor how to adapt the verifier reward and evaluation protocols to measure intensity-aware compliance; extending **V-RoLoRA** to integer-valued control vectors remains an important avenue for future work.

References

Jadie Adams, Brian Hu, Emily Veenhuis, David Joy, Bharadwaj Ravichandran, Aaron Bray, Anthony Hoogs, and Arslan Basharat. 2025. Steerable pluralism: Pluralistic alignment via few-shot comparative regression. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(1):15–25.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. 2025. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *ACM Comput. Surv.*, 58(2).

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

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning {ai} with shared human values. In *International Conference on Learning Representations*.

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

Liwei Jiang, Taylor Sorensen, Sydney Levine, and Yejin Choi. 2025. Can language models reason about individualistic human values and preferences? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6757–6794, Vienna, Austria. Association for Computational Linguistics.

Dongjun Kang, Joonsuk Park, Yohan Jo, and JinYeong Bak. 2023. From values to opinions: Predicting human behaviors and stances using value-injected large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15539–15559, Singapore. Association for Computational Linguistics.

Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. 2022. Identifying the human values behind arguments. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4459–4471, Dublin, Ireland. Association for Computational Linguistics.

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

Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. 2025. Hmora: Making llms more effective with hierarchical mixture of lora experts. In *International Conference on Representation Learning*, volume 2025, pages 91309–91330.

Chi Liu, Derek Li, Yan Shu, Robin Chen, Derek Duan, Teng Fang, and Bryan Dai. 2025. Fleming-r1: Toward expert-level medical reasoning via reinforcement learning. *Preprint*, arXiv:2509.15279.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1104–1114, New York, NY, USA. Association for Computing Machinery.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. 2025. General-reasoner: Advancing llm reasoning across all domains. *Preprint*, arXiv:2505.14652.

Yufei Ma, Zihan Liang, Huangyu Dai, Ben Chen, Dehong Gao, Zhuoran Ran, Wang Zihan, Linbo Jin, Wen Jiang, Guannan Zhang, Xiaoyan Cai, and Libin Yang. 2024. MoDULA: Mixture of domain-specific and universal LoRA for multi-task learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2758–2770, Miami, Florida, USA. Association for Computational Linguistics.

Shalom Schwartz. 2012. An overview of the schwartz theory of basic values. *Online Readings in Psychology and Culture*, 2.

ByteDance Seed, :, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, Yufeng Yuan, Yu Yue, Lin Yan, Qiying Yu, Xiaochen Zuo, Chi Zhang, Ruofei Zhu, Zhecheng An, and 255 others. 2025. Seed1.5-thinking: Advancing superb reasoning models with reinforcement learning. *Preprint*, arXiv:2504.13914.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Taylor Sorensen, Liwei Jiang, Jena D. Hwang, Sydney Levine, Valentina Pyatkin, Peter West, Nouha Dziri, Ximing Lu, Kavel Rao, Chandra Bhagavatula, Maarten Sap, John Tasioulas, and Yejin Choi. 2024a. Value kaleidoscope: Engaging ai with pluralistic human values, rights, and duties. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18):19937–19947.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024b. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46280–46302. PMLR.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Chen Wang, Lai Wei, Yanzhi Zhang, Chenyang Shao, Zedong Dan, Weiran Huang, Yuzhi Zhang, and Yue Wang. 2025a. Eframe: Deeper reasoning via exploration-filter-replay reinforcement learning framework. *Preprint*, arXiv:2506.22200.

Xujia Wang, Haiyan Zhao, Shuo Wang, Hanqing Wang, and Zhiyuan Liu. 2025b. MALoRA: Mixture of asymmetric low-rank adaptation for enhanced multi-task learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5609–5626, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhangchen Xu, Yuetai Li, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. 2025. Tinyv: Reducing false negatives in verification improves rl for llm reasoning. *Preprint*, arXiv:2505.14625.

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

Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Weizhu Chen, and Yunhai Tong. 2025b. Mtl-lora: Low-rank adaptation for multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20):22010–22018.

Jing Yao, Xiaoyuan Yi, Shitong Duan, Jindong Wang, Yuzhuo Bai, Muhua Huang, Yang Ou, Scarlett Li, Peng Zhang, Tun Lu, Zhicheng Dou, Maosong Sun, James Evans, and Xing Xie. 2025. Value compass benchmarks: A comprehensive, generative and self-evolving platform for LLMs' value evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 666–678, Vienna, Austria. Association for Computational Linguistics.

Jing Yao, Xiaoyuan Yi, Yifan Gong, Xiting Wang, and Xing Xie. 2024. Value FULCRA: Mapping large language models to the multidimensional spectrum

of basic human value. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8762–8785, Mexico City, Mexico. Association for Computational Linguistics.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yunfan Zhang, Kathleen McKeown, and Smaranda Muresan. 2025. Exploring chain-of-thought reasoning for steerable pluralistic alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25647–25660, Suzhou, China. Association for Computational Linguistics.

Yiyun Zhou, Chang Yao, and Jingyuan Chen. 2025. CoLA: Collaborative low-rank adaptation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14115–14130, Vienna, Austria. Association for Computational Linguistics.

Caleb Ziems, Jane Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. 2022. The moral integrity corpus: A benchmark for ethical dialogue systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3755–3773, Dublin, Ireland. Association for Computational Linguistics.

Heming Zou, Yunliang Zang, Wutong Xu, Yao Zhu, and Xiangyang Ji. 2025. FlyloRA: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Notation Summary

- x: input prompt; y: generated response; $\pi_\theta(y \mid x, v)$: conditional generation policy.
- $v \in \{0,1\}^K$: discrete condition vector with K dimensions.
- W (or W_0): frozen pretrained weight matrix of the underlying linear layer; ΔW: LoRA-style update.
- $A \in \mathbb{R}^{d_{\text{in}} \times r}$, $B \in \mathbb{R}^{r \times d_{\text{out}}}$: LoRA low-rank factors; r: LoRA rank; α: LoRA scaling factor.
- M: number of LoRA experts; $(A^{(m)}, B^{(m)})$: the m-th LoRA expert parameters; $g(v) \in \mathbb{R}^M$: expert mixture weights with $\sum_{m=1}^{M} g_m(v) = 1$.
- $E \in \mathbb{R}^{V \times K}$: frozen sparse Gaussian projection matrix; V: projection (routing) feature dimension.
- W_g and b: trainable gate parameters; $z = W_g E v^T + b$: routing logits; $g(v) = \mathrm{softmax}(z)$.
- $\mathcal{E}_\phi$: external discriminator (model-based verifier); $\hat{v}(x, y) = \mathcal{E}_\phi(x, y) \in \{0,1\}^K$: predicted condition.
- $\mathcal{C}(v, \hat{v}) \in [0,1]$: conditional consistency score (e.g., normalized multi-label accuracy); $\mathcal{O}(x, y)$: auxiliary objective score.
- $r(x, v, y)$: scalar reward; $\lambda_{\text{cons}}, \lambda_{\text{other}}$: reward weights.
- G: GRPO group size (number of sampled responses per (x, v)); $\bar{r}_G(x, v)$: group baseline; $A(x, v, y_i)$ and $\hat{A}(x, v, y_i)$: (normalized) advantages; ϵ: numerical stability constant.

B Datasets

We use two existing datasets in our work: **Touché23-ValueEval**, and **MIC(Moral Integrity Corpus)**. More information about the source datasets and their licenses is provided below.

- **Touché23-ValueEval** (Schwartz, 2012): The dataset is released under the CC-BY-4.0 License and is publicly available at `https://huggingface.co/datasets/webis/Touche23-ValueEval`.
- **MIC(Moral Integrity Corpus)** (Ziems et al., 2022): The dataset is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License and is publicly available at `https://github.com/SALT-NLP/mic`.

All source datasets used in our benchmark retain their original licenses, as specified by their respective creators.

B.1 Value/Moral Systems

Schwartz's Basic Human Values We leverage the widely used Schwartz Theory of Basic Values(Schwartz, 2012) to define value profiles for the Touché23-ValueEval dataset. This theory proposes ten universal values that reflect broad human motivations and are organized in a circular continuum to indicate their compatibilities and conflicts. These ten values are grouped into four higher-order

Table 3: Definition of 10 value dimensions in Schwartz's Theory of Basic Values.

Value Dimension	Value Definition
Power	Pursuit of social status, control over resources, and dominance over others.
Achievement	Personal success demonstrated through competence according to social standards.
Hedonism	Seeking pleasure, sensory gratification, and enjoyment in life.
Stimulation	Desire for novelty, excitement, and challenging experiences.
Self-Direction	Independent thought, freedom of choice, creativity, and exploration.
Universalism	Understanding, tolerance, and protection for all people and nature.
Benevolence	Commitment to the welfare of close others—emphasizing care and loyalty.
Tradition	Respect and acceptance of cultural or religion customs.
Conformity	Restraint of actions that may upset or harm others or violate social norms.
Security	Pursuit of safety, harmony, and societal or personal stability.

meta-types: Openness to Change, Conservation, Self-Enhancement, and Self-Transcendence. Table 3 summarizes these ten value dimensions and their motivational definitions.

For the MIC dataset, we follow the moral taxonomy adopted in the *Moral Integrity Corpus (MIC)* (Ziems et al., 2022) to define moral profiles. MIC is grounded in *Moral Foundations Theory*, which models human moral judgments through a small set of foundational motivations that capture why an utterance or behavior may be perceived as ethically acceptable or problematic. Concretely, MIC operationalizes morality with six foundations—CARE, FAIRNESS, LIBERTY, LOYALTY, AUTHORITY, and SANCTITY—each paired with an opposing vice (e.g., Harm, Cheating, Oppression, Betrayal, Subversion, and Degradation). In MIC's annotation protocol, each *Rule of Thumb* (RoT) can be assigned one or more foundations, enabling multi-label evaluation of a model's moral consistency across diverse dialogue situations (Table 4).

B.2 Evaluation Metrics

To quantitatively assess how well a generated response aligns with a target multi-label condition vector, we compare the predicted label set to the ground-truth labels for each example. Suppose the dataset contains N evaluation instances. For the n-th instance, let the ground-truth binary vector be $y^{(n)} \in \{0,1\}^K$ and the predicted binary vector be $\hat{y}^{(n)} \in \{0,1\}^K$, where K is the number of value/moral dimensions (e.g., K=10 for valueEval and K=6 for MIC). We use three complementary metrics: **micro-F1**, **macro-F1**, and **Jaccard**.

Micro-F1. Micro-F1 aggregates true/false positives and negatives over all labels and examples, emphasizing overall decision quality and being more sensitive to frequent labels. Define the global counts:

$$\mathrm{TP} = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}\left[\hat{y}_k^{(n)} = 1 \wedge y_k^{(n)} = 1\right], \quad (15)$$

$$\mathrm{FP} = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}\left[\hat{y}_k^{(n)} = 1 \wedge y_k^{(n)} = 0\right], \quad (16)$$

$$\mathrm{FN} = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}\left[\hat{y}_k^{(n)} = 0 \wedge y_k^{(n)} = 1\right]. \quad (17)$$

Micro-precision and micro-recall are:

$$P_\mu = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad R_\mu = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad (18)$$

and micro-F1 is the harmonic mean:

$$\text{Micro-F1} = \frac{2P_\mu R_\mu}{P_\mu + R_\mu} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}. \quad \text{(M1)}$$

Macro-F1. Macro-F1 computes F1 for each label dimension independently and then averages across dimensions, thus highlighting balanced performance on both head and tail labels. For each label k, define:

$$\mathrm{TP}_k = \sum_{n=1}^{N} \mathbb{I}\left[\hat{y}_k^{(n)} = 1 \wedge y_k^{(n)} = 1\right], \quad (19)$$

$$\mathrm{FP}_k = \sum_{n=1}^{N} \mathbb{I}\left[\hat{y}_k^{(n)} = 1 \wedge y_k^{(n)} = 0\right], \quad (20)$$

$$\mathrm{FN}_k = \sum_{n=1}^{N} \mathbb{I}\left[\hat{y}_k^{(n)} = 0 \wedge y_k^{(n)} = 1\right]. \quad (21)$$

Table 4: Definition of 6 moral dimensions in the Moral Integrity Corpus (MIC).

Moral Dimension	Moral Definition
Care	Wanting someone or something to be safe, healthy, and happy.
Fairness	Wanting to see individuals or groups treated equally or equitably.
Liberty	Wanting people to be free to make their own decisions.
Loyalty	Wanting unity and seeing people keep promises or obligations to an in-group.
Authority	Wanting to respect social roles, duties, privacy, peace, and order.
Sanctity	Wanting people and things to be clean, pure, innocent, and holy.

Then the per-label precision, recall, and F1 are:

$$P_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k}, \tag{22}$$

$$R_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k}, \tag{23}$$

$$F1_k = \frac{2P_kR_k}{P_k + R_k}, \tag{24}$$

and macro-F1 is:

$$\text{Macro-F1} = \frac{1}{K}\sum_{k=1}^{K} F1_k. \tag{M2}$$

Jaccard (example-level set overlap). Jaccard directly measures set-level overlap between predicted and gold label sets, reflecting consistency under compositional multi-label combinations. For the n-th instance, let

$$Y^{(n)} = \{k \mid y_k^{(n)} = 1\}, \qquad \hat{Y}^{(n)} = \{k \mid \hat{y}_k^{(n)} = 1\}. \tag{25}$$

The example-level Jaccard is:

$$J^{(n)} = \frac{\left|Y^{(n)} \cap \hat{Y}^{(n)}\right|}{\left|Y^{(n)} \cup \hat{Y}^{(n)}\right|}, \tag{M3}$$

and we report the mean Jaccard over all instances:

$$\text{Jaccard} = \frac{1}{N}\sum_{n=1}^{N} J^{(n)}. \tag{M4}$$

Together, these metrics provide a holistic view of controllability: micro-F1 captures overall correctness aggregated across labels, macro-F1 emphasizes balanced per-dimension performance, and Jaccard evaluates compositional set-level agreement between the target and predicted value/moral profiles.

C Baselines

We compare Conditional MOELoRA with a total of seven representative PEFT baselines on both Qwen and Llama backbones. For all LoRA-related methods, we apply adapters to all linear modules (down_proj, k_proj, v_proj, q_proj, up_proj, gate_proj, o_proj) for a fair comparison.

C.1 Method Descriptions

- **LoRA:** LoRA is a low-rank decomposition technique that reduces trainable parameters by inserting trainable low-rank updates into the original model weights.
- **HydraLoRA:** HydraLoRA is an asymmetric fine-tuning architecture that allocates distinct up-projection matrices for task-specific features while sharing a down-projection matrix to capture global information.
- **MOELoRA:** MOELoRA replaces a single LoRA adapter with a pool of LoRA experts and aggregates them through a router to better handle task heterogeneity and data imbalance.
- **MTL-LoRA:** MTL-LoRA enhances LoRA with task-conditioned parameters to improve multi-task transfer while keeping the number of trainable parameters small.
- **CoLA:** CoLA generalizes LoRA by introducing a many-to-many collaboration between multiple A and B matrices, enabling more flexible low-rank composition strategies.
- **MALoRA:** MALoRA (Mixture of Asymmetric Low-Rank Adaptation) leverages asymmetry between down- and up-projection modules by sharing a trainable low-rank subspace for down-projection while assigning each expert a compact coefficient matrix, improving efficiency.

Method	Hyperparameter	Setting
LoRA	r	8, 16, 24, 32, 64
	lora_alpha	32
CoLA	r	8
	#A | #B	1|3, 2|3, 1|14, 4|10
MOELoRA	r	32
	lora_alpha	32
	expert_num	8
	task_embedding_dim	64
MALoRA	r	8
	lora_alpha	32
	expert_num	8
	down_rank	4
	task_embedding_dim	64
MTL-LoRA	r	8
	lora_alpha	32
	expert_num	8
	task_embedding_dim	64
FlyLoRA	r	8
	lora_alpha	32
	expert_num	8
	task_embedding_dim	64
HydraLoRA	r	8
	lora_alpha	32
	expert_num	8
	task_embedding_dim	64

Table 5: Hyperparameter settings for PEFT baselines.

- **FlyLoRA:** FlyLoRA is an implicit MoE-style LoRA variant that mitigates interference via rank-wise expert activation and an implicit router based on a frozen sparse random projection.

C.2 Hyperparameter Settings

Table 5 summarizes the hyperparameters used for each baseline. Unless otherwise stated, we set `lora_alpha`$= 32$ and tune the LoRA rank r as specified. For CoLA, we report the number of collaborating A and B matrices (#A|#B). For MoE-style baselines, we use the same number of experts and task-embedding dimensionality for fairness.

D Human Evaluation

To complement automatic evaluation, we conduct human judgments on how well model-generated responses align with the target value/moral profiles on both Touché23-ValueEval and MIC. For Touché23-ValueEval (Kang et al., 2023), we sample 200 instances from the evaluation split, covering four representative value profiles provided by the benchmark (two country-level profiles and two group-level profiles), with 50 instances per profile. For MIC (Ziems et al., 2022), we sample 200 instances and stratify them by target moral-profile combinations (defined over the six moral dimensions) to ensure coverage of both single-dimension and multi-dimension targets.

Annotation protocol. Each evaluation instance presents annotators with (i) a prompt question, (ii) a target profile, and (iii) two candidate responses (Answer A and Answer B) produced by two different models. For Touché23-ValueEval, the target profile specifies the desired value orientations (a subset of Schwartz values) along with short descriptions (and priority information when provided). For MIC, the target profile specifies the desired moral dimensions (a subset of Care/Fairness/Liberty/Loyalty/Authority/Sanctity) with brief definitions. Annotators are instructed to first identify which values/moral dimensions are reflected by each response, and then decide which response better matches the target profile. They choose one of three options: WIN (Answer A better matches the target profile), TIE (both are comparably aligned), or LOSE (Answer B better matches). The order of Answer A/B is randomized and model identities are hidden to mitigate position and branding biases. Each instance is independently evaluated by three annotators, and the final label is obtained via majority voting. Example questionnaires for Touché23-ValueEval and MIC are shown in Table 6 and Table 7, respectively.

Annotators and ethics. We recruited three annotators through a professional vendor. All annotators have a background in psychology or related social-science fields, are proficient in English reading comprehension, and underwent training on Schwartz's theory of basic human values and the moral-dimension definitions used in MIC. Annotators were compensated at an hourly rate of $7. All annotation procedures were approved by the relevant Institutional Review Board (IRB) and conducted in accordance with ethical research guidelines. Annotators were informed about the task scope and voluntarily consented to participate. No personally identifiable information was collected during the study.

E External Discriminator Details

To compute the conditional consistency reward in Section 3.3, we train two lightweight *external discriminators* based on `Qwen/Qwen3-0.6B-Base`

Statement/Question	Answer A	Answer B	Target Values	Label
Tell me what you would say about the following statement. **Statement:** We should abandon the use of school uniform.	**Stance: Yes** I would say, "I am in favor of abandoning school uniforms because they are not necessary and can be a financial burden for some parents."	**Stance: No** I would say, "I am against abandoning school uniforms because they help promote unity and a shared identity within the school."	Security (safety, harmony, and stability of society and relationships) Universalism (understanding, appreciation, tolerance, and protection for the welfare of all people and for nature)	Win / Tie / Lose

Table 6: An example questionnaire item for human evaluation on Touché23-ValueEval.

Question	Answer A	Answer B	Target Morals	Label
Prompt: Which response better matches the target moral profile? **Q:** I took a picture of a friend's daughter (age 3) while I was babysitting. It's a really cute pic that I printed with the intention of giving the only copy to my friend/the kid's mom. Is that OK or creepy?	I think it's totally fine. I don't think it's creepy at all.	I don't think it's creepy. It's just a picture of a cute kid.	Care (wanting someone or something to be safe, healthy, and happy) Authority (wanting to respect social roles, duties, privacy, peace, and order) Sanctity (wanting to live in a way that is clean, pure, and holy)	Win / Tie / Lose

Table 7: An example questionnaire item for human evaluation on MIC.

using the `transformers`[4] implementation `Qwen3ForSequenceClassification`. One discriminator is trained on VALUEEVAL to predict the K=10-dimensional value vector, and the other is trained on MIC to predict the K=6-dimensional moral vector. Given an input prompt x and a candidate response y, we build the discriminator input by concatenating (x, y) into a single sequence (truncated/padded to a maximum length), and obtain a pooled representation $h(x, y)$ from the backbone encoder. The classifier then produces multi-label logits:

$$s(x, y) = Wh(x, y) + b \in \mathbb{R}^K, \qquad (26)$$

and converts them into per-dimension probabilities via a sigmoid:

$$p(x, y) = \sigma\big(s(x, y)\big). \qquad (27)$$

We derive the binary prediction $\hat{v}(x, y) \in \{0, 1\}^K$ using a fixed threshold $\tau = 0.5$:

$$\hat{v}_k(x, y) = \mathbb{I}\big[p_k(x, y) \geq \tau\big], \quad k = 1, \ldots, K. \qquad (28)$$

During training, we optimize a multi-label binary cross-entropy objective (BCE with logits) against the ground-truth label vector (values for VALUEEVAL or morals for MIC). To reduce computational cost while retaining classification quality, we fine-tune the discriminators with LoRA adapters.

[4] https://github.com/huggingface/transformers

Hyperparameter	Setting
Backbone model	`Qwen/Qwen3-0.6B-Base`
Classifier implementation	`Qwen3ForSequenceClassification`
Tasks	VALUEEVAL (K=10), MIC (K=6)
Loss	multi-label BCE (with logits)
Max sequence length	768
Learning rate	2×10^{-4}
Training epochs	3
Train batch size (per device)	4
Eval batch size (per device)	4
Gradient accumulation steps	1
Weight decay	0.01
Warmup ratio	0.03
LoRA rank r	8
LoRA scaling α	16
LoRA dropout	0.05
Threshold τ	0.5
Logging steps	50
Evaluation steps	200
Checkpoint saving steps	200
Random seed	42

Table 8: Training configuration of the external discriminators for VALUEEVAL and MIC. Unless otherwise noted, the same hyperparameters are used for both discriminators.

F Experimental Details

This appendix summarizes additional implementation details, including (i) how we inject conditions for prompt-based baselines and (ii) the hyperparam-

eter settings for fine-tuning and GRPO. Tables 9–11 provide the full configurations.

F.1 Prompt-based Condition Injection for LoRA/CoLA

For prompt-based baselines (LoRA and CoLA), we inject the target condition via a *system-style condition prompt* prepended to the original input. We use the same template for both training and evaluation. When writing the templates in the main paper, we replace the concrete value/moral names and their descriptions with symbolic placeholders.

• Condition Prompt Template for Touché23-ValueEval

System: You are a person whose core values include: `{value_list}`. Your opinions and arguments should be consistent with these values.
Here are the values you strongly endorse:
- `{value_1}`: `{definition_1}`
- `{value_2}`: `{definition_2}`
...
- `{value_k}`: `{definition_k}`
When you respond, speak as someone who highly values the points listed above.

• Condition Prompt Template for MIC

System: You are a person whose core moral principles include: `{moral_list}`. Your answers to questions should clearly reflect and embody these moral principles.
Here are the morals you strongly endorse:
- `{moral_1}`: `{definition_1}`
...
- `{moral_k}`: `{definition_k}`

F.2 Hyperparameter Settings

Fine-tuning configuration. Table 9 reports our default supervised fine-tuning (cold-start) configuration. Unless otherwise stated, these settings are shared across methods; method-specific adapter hyperparameters are listed separately in Table 10.

Method-specific adapter hyperparameters. Table 10 summarizes the key adapter hyperparameters for each PEFT method. For LoRA, we sweep the rank r; for CoLA, we additionally vary the collaboration structure (#A|#B). For MoE-style baselines, we follow the settings below for a fair comparison.

Hyperparameter	Setting
Per-device batch size	4
Gradient accumulation steps	8
Total update steps	3000
Learning rate	9×10^{-6}
LoRA dropout	0.1
Bias_u	0.01
Scheduler	cosine
Optimizer	AdamW
Hardware	two NVIDIA A100-SXM4 (80GB) GPUs

Table 9: Default hyperparameter settings for the cold-start supervised fine-tuning stage.

Method	Hyperparameter	Setting
MOELoRA	r	32
	`lora_alpha`	32
	`expert_num`	8
	`task_embedding_dim`	64
LoRA	r	8, 16, 24, 32, 64
	`lora_alpha`	32
HydraLoRA	#A | #B	1 | 3, 1 | 14
MALoRA	`down_rank`	4
MTL-LoRA	#A | #B	1 | 14
FlyLoRA	r	8
CoLA	r	8
	#A | #B	1 | 3, 2 | 3, 1 | 14, 4 | 10

Table 10: Method-specific adapter hyperparameters used in our experiments.

GRPO (RLVR) hyperparameters. Table 11 lists the GRPO configuration used in our RLVR stage (Section 3.3). We sample $G =$ `num_generations` candidates per (x, v) and apply group-relative normalization in the GRPO objective.

GRPO Hyperparameter	Setting
num_generations (G)	8
max_completion_length	200
temperature	1.0
top_p	1.0
top_k	0
repetition_penalty	1.0
beta	0.0
num_iterations	1
epsilon	0.2
sapo_temperature_neg	1.05
sapo_temperature_pos	1.0
ref_model_mixup_alpha	0.6
ref_model_sync_steps	512
top_entropy_quantile	1.0

Table 11: GRPO hyperparameters for the RLVR stage.