MULTI-LLM COLLABORATIVE ALIGNMENT VIA STACKELBERG GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

A pool of language models can collaborate and improve collectively by learning from one another's responses. These interactions depend on the instructions used during training. Existing methods typically sample instructions uniformly, even though their usefulness may change as the models improve: an instruction on which models' responses once differed in quality may later be answered equally well, while a previously difficult instruction may begin to provide a useful learning signal. We propose STACKELBERG ALIGNMENT, a game-theory-inspired leader–follower framework that turns instruction selection into an adaptive curriculum. An EXP3 bandit acts as the leader, allocating a fixed sampling budget across instructions and updating its sampling distribution using a reward that combines instruction difficulty and response discriminability. The language models act as followers: they respond to the selected instructions, evaluate one another's responses, and learn from the resulting preference signals through DPO or GRPO. The framework uses Elo-style reputation-weighted peer judgment and reputation-based opponent matching to support reliable and competitive model interactions. Experiments across three heterogeneous model pools and 12 benchmarks spanning scientific discovery, reasoning, code, instruction following, and knowledge show that STACKELBERG ALIGNMENT achieves the highest macro-average across three diverse model pools, outperforming the strongest training-time baseline by up to 7.4% and the best static inference baseline by 12–25%. Analysis confirms that the adaptive leader concentrates duels on the most informative instructions, and ablations show that both reputation-weighted judgment and reputation-based matching improve the effectiveness of multi-LLM evolution.

1 INTRODUCTION

Model collaboration allows language models to benefit from one another's complementary capabilities, and through inference-time interactions produce outputs better than any individual model (Du & Kaelbling, 2024; Feng et al., 2026a). Training-time collaboration offers a more fundamental opportunity: models can improve by interacting with an environment of other models, generating training signals that neither could produce alone. Recent methods often achieve this through comparison and competition (Luo et al., 2024; Subramaniam et al., 2025). Sparta Alignment (Jiang et al., 2025), for example, has pairs of models respond to the same instruction, the remaining models judge the responses, and the winning pair is used for preference learning, reliably lifting all models in the pool across reasoning, code, and open-ended tasks. It further maintains reputation scores that weight peer judgments and guide opponent matching; we build on top of this framework. A key ingredient to all of these methods is, however, left unexplored: *which instructions should models compete on*. Existing approaches sample instructions uniformly throughout training with fixed schedule, rather than treating instruction selection as an adaptive component of the training loop.

Uniform instruction sampling is inefficient because instruction informativeness is non-stationary and heterogeneous. Instructions where all models already agree yield no discriminative preference signal; instructions that all models fail on produce only tied scores with no learning gradient; the most valuable instructions are those at the *frontier of disagreement*, where duels and pairwise comparisons generate learnable preference pairs. This frontier shifts as models improve across iterations: an instruction that drives rich learning early in training may become trivial later, while harder instructions become tractable only after the pool has strengthened. A fixed sampling distribution cannot track

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

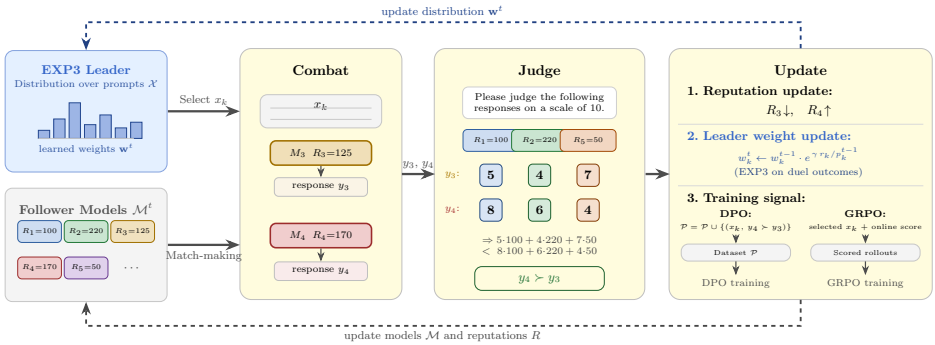


Figure 1: Overview of STACKELBERG ALIGNMENT for one iteration. The EXP3 leader maintains learned weights $\mathbf{w}^t$ over the instruction pool and selects instruction x_k non-uniformly. Two follower models (M_3, M_4 in the figure) are matched by reputation and duel on x_k, generating responses y_3 and y_4. Remaining models act as peer judges, scoring both responses; scores are aggregated weighted by judge reputation, determining the winner. The Update step performs three actions: (1) reputation scores are adjusted, (2) leader weights are updated via EXP3 based on duel informativeness, and (3) the winning preference pair is added to the dataset $\mathcal{P}$ for DPO training, or the selected instruction and online judge scores are used directly as the reward for GRPO training. Dashed arrows show the resulting feedback loop: the updated leader weights reshape the instruction distribution, and the updated reputations and retrained models feed back into the follower pool for the next iteration.

these shifts, wasting a large fraction of compute on uninformative instructions. Adaptive instruction selection, informed by the evolving capabilities of the model pool, is therefore central to efficient collaborative alignment and training-time collaboration in general.

We frame this observation as a bilevel Stackelberg game (von Stackelberg & Peacock, 1952; Başar & Olsder, 1998) between two classes of players. A **leader** controls a distribution over the instruction pool and commits to it before any model generates a response; its objective is to concentrate compute on instructions that currently lie at the frontier of model disagreement. The **followers** are the LLMs in the pool: they observe the selected instruction, duel, and are trained on the resulting preference pairs. The leader implements this via EXP3 (Auer et al., 2002), a no-regret bandit algorithm well suited to non-stationary reward sequences, with a composite reward that captures both instruction difficulty and duel discriminability. Each model also maintains a reputation score reflecting its track record in past duels; models not part of a duel act as peer judges whose votes are weighted by their own reputations, and opponents are matched by reputation proximity so that duels remain competitive as skill levels diverge across training. The resulting method, STACKELBERG ALIGNMENT, is the first instantiation of collaborative LLM alignment with an adaptive leader that learns the instruction distribution jointly with the follower models, yielding a dynamic curriculum that concentrates compute on instructions where the pool still disagrees and shifts automatically as models strengthen.

We evaluate STACKELBERG ALIGNMENT on three pools of heterogeneous LLMs spanning specialized expert models, diverse academic research models, and general-purpose models, across 12 benchmarks covering scientific discovery, reasoning, code, instruction following, and knowledge. STACKELBERG ALIGNMENT achieves the highest macro-average in all three pools, outperforming other training-time algorithms by up to 7.4% and the best static inference baseline by 12–25%. Our main contributions are: (1) a Stackelberg game formulation of training-time multi-LLM collaborative alignment; (2) an EXP3 bandit leader with a composite duel-informativeness reward that tracks the shifting capability frontier of the model pool; and (3) an empirical demonstration of consistent gains across three diverse model pools and twelve benchmarks, with analysis tracing the gains to the adaptive curriculum.

2 METHOD

Figure 1 illustrates one complete training iteration of STACKELBERG ALIGNMENT.

Setup. Let $\mathcal{M}^0 = \{M_1^0, \ldots, M_m^0\}$ be a pool of m language models and $\mathcal{X} = \{x_1, \ldots, x_K\}$ a dataset of instructions. Our goal is to produce an improved pool $\mathcal{M}^T$ by having models interact, evaluate, and learn from each other across T iterations. Models engage in pairwise *duels* on instructions: two models generate responses, the remaining models act as peer judges and score each

response on a $[s_{\min}, s_{\max}]$ scale, and these scores are aggregated into preference pairs for RL fine-tuning. Each iteration runs D such duels, with the instruction for each duel drawn independently (with replacement) from the leader's distribution π_L rather than sweeping the instruction pool exhaustively; a given instruction may therefore be drawn multiple times, once, or not at all within a single iteration, and only duels that are actually drawn contribute preference pairs or reward signal to that iteration's training. This is an intentional design choice rather than an incidental side effect: concentrating duels on the currently most informative instructions, instead of exhausting the pool every iteration, is exactly the adaptive curriculum this framework is designed to realize (§2.1). Because the leader's sampling distribution keeps a nonzero floor probability for every instruction (Eq. 1), no instruction is permanently excluded, though there is no guarantee that every instruction in the pool is drawn within any single iteration or even over the full training run. Following Jiang et al. (2025), each model M_i carries a *reputation score* $R_i \in \mathbb{R}$ and a *reputation deviation* σ_i, initialized uniformly. Reputation encodes each model's perceived reliability, as judged by its peers over training, and serves two roles: it governs opponent selection, and weights peer judgments during score aggregation.

A Stackelberg view of training-time model collaboration. Existing model collaboration approaches (Du et al., 2024; Subramaniam et al., 2025; Wang et al., 2025a; Jiang et al., 2025) treat instruction selection as fixed, sampling uniformly from the training pool regardless of what the current model pool still needs to learn. However, instructions vary widely in difficulty and informativeness: some are already mastered by all models and yield no useful gradient, others are beyond reach and produce no meaningful preference signal, and the most valuable ones lie at the frontier of disagreement where duels generate genuine learning signal. Critically, as models collaborate and improve across iterations, this frontier shifts, making the optimal instruction distribution non-stationary. An adaptive curriculum that tracks these shifts is therefore essential for efficient collaborative alignment. We frame the duel-based training loop above as a **bilevel Stackelberg game** (von Stackelberg & Peacock, 1952) between two classes of players.

The **leader** L controls a distribution $\pi_L \in \Delta(\mathcal{X})$ over the instruction pool and commits to it before any model generates a response. Its strategic objective is to select the instructions that produce the most informative learning signal. The **followers** $\{M_i\}_{i=1}^m$ observe the selected instruction $x \sim \pi_L$ and respond by generating outputs, which are then evaluated and used for training. This sequential commitment structure (leader first, followers second) defines a Stackelberg game (Başar & Olsder, 1998), and the leader's optimal strategy anticipates the followers' most informative responses.

This framework subsumes several prior alignment paradigms as special cases. When π_L is fixed to uniform and $m = 1$, it recovers single-model self-alignment methods such as Self-Rewarding (Yuan et al., 2025) and SPIN (Chen et al., 2024). When π_L is uniform but $m > 1$ with pairwise competition, it recovers SPARTA ALIGNMENT (Jiang et al., 2025): we adopt its reputation-weighted peer judgment and Elo-style reputation updates directly (§2.3), and extend its reputation-based opponent matching with a time-varying gap schedule (§2.2). STACKELBERG ALIGNMENT is the first instantiation of the full framework with an *adaptive* leader that learns π_L jointly with the followers.

Algorithm 1 summarizes the full procedure; the subsections below detail each component.

2.1 STACKELBERG LEADER: ADAPTIVE INSTRUCTION SELECTION

Instruction sampling distribution. At each training iteration $t \in \{1, \ldots, T\}$, the leader must pick which of the K candidate instructions $\mathcal{X} = \{x_k\}_{k=1}^K$ to use for that iteration's duels; p_k^t denotes the probability of sampling instruction x_k at iteration t, and instructions that recently produced informative duels should be sampled more often. To this end, the leader maintains a weight vector $\mathbf{w}^t = (w_1^t, \ldots, w_K^t) \in \mathbb{R}_{>0}^K$ and induces the sampling distribution p_k^t via EXP3 (Auer et al., 2002):

$$p_k^t = (1 - \gamma) \frac{w_k^t}{\|\mathbf{w}^t\|_1} + \frac{\gamma}{K}, \tag{1}$$

where $\gamma \in [0, 1]$ is the exploration rate. The $(1 - \gamma)$ term exploits high-weight instructions while the γ/K term encourages that all instructions are visited. EXP3 is fitting here because instruction rewards are non-stationary, meaning an instruction's informativeness is not fixed but drifts over the course of training: as models improve across iterations, the same instruction may yield rich signal early and none later. Many bandit algorithms assume each instruction's reward is drawn from a fixed distribution and can perform poorly once that assumption breaks; EXP3 instead offers a no-regret

Algorithm 1 STACKELBERG ALIGNMENT

Require: Model pool $\mathcal{M}^0$, instruction set $\mathcal{X} = \{x_k\}_{k=1}^K$, iterations T, duels per iteration D
Initialize reputation $R_i \leftarrow R_0$, $\sigma_i \leftarrow \sigma_0$ for all i; leader weights $\mathbf{w}^0 \leftarrow \mathbf{1}_K$
for $t = 1, \ldots, T$ **do**
  Judged duels $\mathcal{J} \leftarrow \emptyset$; preference pairs $\mathcal{P} \leftarrow \emptyset$
  for each duel $d = 1, \ldots, D$ **do**
    $x \sim \pi_L(\mathbf{w}^{t-1})$ ▷ Leader commits to instruction
    Select active model M_i^t; draw opponent $M_{i'}^t \sim p_{\text{match}}(\cdot \mid M_i^t, \{R_j\}, t)$ ▷ Eq. 7
    Combat: generate $y_i \leftarrow M_i^t(x)$, $y_{i'} \leftarrow M_{i'}^t(x)$
    Judge: each non-combatant M_k scores responses: $s_i^{(k)}, s_{i'}^{(k)} \in [s_{\min}, s_{\max}]$ ▷ §2.3
    $\bar{s}_i, \bar{s}_{i'} \leftarrow \text{Aggregate}\big(\{s_i^{(k)}, s_{i'}^{(k)}\}_{k \notin \{i,i'\}}, \{R_k\}\big)$ ▷ Eq. 8
    Add $(x, y_i, y_{i'}, \bar{s}_i, \bar{s}_{i'})$ to $\mathcal{J}$; update $R_i, R_{i'}, \sigma_i, \sigma_{i'}$ ▷ Eq. 9; see Appx. C
    If $\bar{s}_i \neq \bar{s}_{i'}$: add preference pair to $\mathcal{P}$
  end for
  Leader update: $\mathbf{w}^t \leftarrow \text{EXP3-Update}(\mathbf{w}^{t-1}, \mathcal{J})$ ▷ uses all judged duels
  Train (DPO): $M_i^t \leftarrow \text{DPO}(M_i^{t-1}, \mathcal{P})$ for all i
  or Train (GRPO): $M_i^t \leftarrow \text{GRPO}(M_i^{t-1}, \{(x, y_i, y_{i'})\} \in \mathcal{J}, \{R_k\})$ for all i ▷ online peer rewards
end for
return Improved model pool $\mathcal{M}^T$

guarantee, i.e., its cumulative reward stays close to that of the best fixed instruction in hindsight, even against such adversarially shifting reward sequences.

Leader reward. After a duel on instruction x_k with peer-aggregated scores $\bar{s}_i, \bar{s}_{i'} \in [s_{\min}, s_{\max}]$ (Eq. 8), the leader receives a composite reward $r_k \in [0, 1]$, which is fed into the EXP3 weight update below to raise or lower how often instruction x_k is sampled in future iterations:

(i) Difficulty reward. Among instructions that are learnable (score above a floor threshold), the leader prefers harder ones: instructions where models score perfectly are already mastered and offer no room for improvement, while those just above the threshold are most informative. For a model's aggregate score s this is:

$$r_{\text{diff}}(s) = \begin{cases} \dfrac{s_{\max} - s}{s_{\max} - \tau_{\text{th}}} & s \geq \tau_{\text{th}} \\ 0 & s < \tau_{\text{th}} \end{cases} \tag{2}$$

where τ_{th} is a minimum score threshold below which the instruction is too hard to yield useful signal.

(ii) Preference quality reward. For preference learning, the gap between chosen and rejected responses matters: deviations from the scheduled target in either direction are penalized: a gap of zero provides no learning signal, and a gap far above the target indicates the comparison is already trivially decided (Yao et al., 2024). We schedule a target gap g_t^* that starts large (clear preference distinctions early) and decays (finer distinctions later):

$$g_t^* = g_{\text{end}} + (g_{\text{start}} - g_{\text{end}})\,(1 - \tau), \qquad \tau = \frac{t}{T - 1}, \tag{3}$$

and reward proximity to this target via a Gaussian kernel:

$$r_{\text{pref}}(\bar{s}_i, \bar{s}_{i'}) = \exp\left(-\frac{\big(|\bar{s}_i - \bar{s}_{i'}| \,/\, (s_{\max} - s_{\min}) - g_t^*\big)^2}{2\sigma_r^2}\right). \tag{4}$$

where σ_r is the kernel bandwidth. The instruction reward is computed per combatant and averaged: $r_k = \frac{1}{2} \sum_{s \in \{\bar{s}_i, \bar{s}_{i'}\}} \big(\lambda_1 r_{\text{diff}}(s) + \lambda_2 r_{\text{pref}}(\bar{s}_i, \bar{s}_{i'})\big)$, with $\lambda_1 + \lambda_2 = 1$.

EXP3 weight update. To correct for the bias introduced by non-uniform sampling, we apply the standard importance-weighted multiplicative update:

$$\log w_k^{t+1} = \log w_k^t + \frac{1}{K} \cdot \frac{r_k \cdot \mathbf{1}[x_k \sim \pi_L^t]}{p_k^t}, \tag{5}$$

followed by re-normalization, where p_k^t is from Eq. 1 and $\mathbf{1}[x_k \sim \pi_L^t]$ indicates whether instruction x_k was sampled in the current duel. Instructions that yield high reward relative to their sampling probability receive increased weight, concentrating future sampling on informative prompts.

2.2 SCHEDULED FOLLOWER MATCHING

For each duel d, the active model M_i^t is selected in round-robin order. Sparta Alignment (Jiang et al., 2025) draws its opponent from $\mathcal{M}^t \setminus \{M_i^t\}$ by reputation proximity; we extend this with a *scheduled soft-matching* policy that targets a time-varying reputation gap, annealing from mismatched to near-peer opponents over training.

Define the normalized reputation gap between M_i^t and any candidate M_j as

$$\hat{g}_{ij}^t = \frac{|R_i^t - R_j^t|}{\max_{k \neq i} |R_i^t - R_k^t|} \in [0, 1]. \tag{6}$$

We select the opponent by sampling from a distribution peaked at a target gap g_t^*:

$$p(M_{i'}^t = M_j) \propto \exp\left(-\frac{(\hat{g}_{ij}^t - g_t^*)^2}{2\sigma_g^2}\right), \quad j \neq i, \tag{7}$$

where σ_g is the selection bandwidth and g_t^* follows Eq. 3 with $g_{\text{start}} = 1$, $g_{\text{end}} = 0$, giving $g_t^* = 1 - \tau$, which linearly decreases from 1 (early: mismatched opponents) to 0 (late: near-peer opponents).

This schedule has a curriculum interpretation. As reputations diverge over training, the early target of $g^* \approx 1$ steers the system toward mismatched opponents: the stronger model nearly always wins, producing clear preference pairs that provide clean supervision for improvement (Sun et al., 2024). Later, as the target shifts toward $g^* \approx 0$, near-peer duels produce finer-grained preference distinctions that drive precise refinement, analogous to competitive matchmaking in human rating systems (Ebtekar & Liu, 2021).

2.3 PEER JUDGMENT AND REPUTATION SYSTEM

Following Jiang et al. (2025), all non-competing models evaluate both combat responses. Each judge $M_k^t \in \mathcal{M}^t \setminus \{M_i^t, M_{i'}^t\}$ assigns scores $s_i^{(k)}, s_{i'}^{(k)} \in [s_{\min}, s_{\max}]$ (a 1–10 scale, i.e., $s_{\min} = 1$, $s_{\max} = 10$) via an LLM-as-a-judge prompt (Zheng et al., 2023). The peer-aggregate score is a reputation-weighted mean:

$$\bar{s}_i = \Big(\sum\nolimits_{k \notin \{i,i'\}} R_k^t \, s_i^{(k)}\Big) \Big/ \Big(\sum\nolimits_{k \notin \{i,i'\}} R_k^t\Big). \tag{8}$$

Weighting by R_k^t discounts the influence of low-reputation judges, so that models with stronger track records contribute more to each score aggregation (Wang et al., 2025b).

Reputation update. Reputations are updated after each duel via an adapted Elo rule (Ebtekar & Liu, 2021), as in Sparta Alignment (Jiang et al., 2025), that incorporates three factors:

$$R_i \leftarrow R_i + (K_t/\lambda) \cdot \underbrace{(\bar{s}_i - \bar{s}_{i'})}_{\text{score gap}} \cdot \underbrace{\tanh(\sigma_i)}_{\text{stability}} \cdot \underbrace{\max\Big(|\Phi(z_i) - \Phi(z_{i'})|,\ \epsilon\Big)}_{\text{gap factor}}, \tag{9}$$

where $z_i = (R_i - R_{i'}) \big/ \sqrt{\sigma_i^2 + \sigma_{i'}^2}$, Φ is the standard normal CDF, $\epsilon > 0$ is a floor preventing stagnation in near-tied duels, and λ is a scaling constant. Reputations are floored at a fixed minimum to prevent degenerate negative drift. The K factor decays exponentially over update count to cool large swings as reputations stabilize; σ_i is the standard deviation of recent reputation deltas over a sliding window, measuring volatility. Full formulas are in Appendix C.

2.4 FOLLOWER TRAINING

STACKELBERG ALIGNMENT supports two training settings for collaborative improvement. For DPO, non-tied preference pairs from all duels are collected and used for offline fine-tuning. For GRPO, the duel assignment structure determines each model's training prompts, with peer judges scoring completions online during training.

DPO. Non-tied preference pairs $(x, y_w \succ y_l)$ collected across all duels are used to fine-tune each model via Direct Preference Optimization (Rafailov et al., 2024), with the previous iteration's checkpoint as the reference policy.

GRPO. As an online alternative, the duel structure determines each model's training prompts. Each model generates G completions per assigned prompt; peer judges score each completion online, and the normalized peer score serves as the per-completion reward in the GRPO objective (Shao et al., 2024). Reputation updates are derived from the logged rewards after training completes.

Connection to the Stackelberg equilibrium. At convergence, the leader's instruction distribution π_L^* maximizes expected instructional informativeness under the followers' best-response policies $\{\pi_i^*\}$; the followers in turn are optimal for the instruction distribution π_L^* generates. This joint fixed point corresponds to a Stackelberg equilibrium (Başar & Olsder, 1998), where neither the leader can improve its choice of instructions given the followers' behavior, nor can the followers improve their policies given the leader's instruction distribution. In practice we approximate this equilibrium via alternating updates: EXP3 for the leader and DPO/GRPO for the followers.

3 Experiment Settings

Models. We employ three model pools of varying scale and architecture for evaluation.

Pool 1 contains 9 heterogeneous models with five Qwen2.5-7B variants (the base instruction-tuned model plus four domain fine-tunes on OASST1, ShareGPT, WizardLM, and biomedical corpora, in Jiang et al. (2025)), alongside AgentFlow-Planner-7B (Li et al., 2026), Llama-3.1-8B-Instruct (Dubey et al., 2024), Aya-Expanse-8B (Dang et al., 2024), and Pangea-7B (Yue et al., 2025). This pool spans diverse model families, training corpora, and areas of expertise.

Pool 2 contains 8 models trained in diverse academic research projects, where researchers contribute specialized models as modular components to a collaborative system. We use the top-8 models sourced in Feng et al. (2026b) (see Appendix D for model identities).

Pool 3 contains 4 general models: Qwen3.5-9B (Qwen Team, 2026), Gemma-4-12B-Instruct (Team et al., 2026), Nemotron-3-Nano-4B (Blakeman et al., 2025), and Phi-4 (Abdin et al., 2024).

Baselines. We compare Stackelberg Alignment in two training configurations, **Stackelberg (DPO)** and **Stackelberg (GRPO)**, against eight baselines spanning static inference and training-based collaboration.

Static inference: **Majority Vote**, ensemble plurality voting (applicable to objective tasks); **MoA** (Wang et al., 2025a), iterative synthesis and aggregation of multiple models' responses; **Multiagent Debate** (Du et al., 2024), iterative response refinement at inference time; **Heterogeneous Swarms** (Feng et al., 2025a), which optimizes directed acylic graphs of model interactions for collaboration. *Training-based*: **Sparta Alignment** (Jiang et al., 2025), the direct predecessor with a uniform instruction leader and DPO-only training; **Multiagent Fine-tuning** (Subramaniam et al., 2025), joint fine-tuning via debate-based data augmentation; **Trained Router** (Ong et al., 2025), which learns to route queries to the best model via supervised fine-tuning on validation-set oracle labels; **AggLM** (Zhao et al., 2025), a GRPO-trained aggregator with verifiable rewards.

Hyperparameters. For Stackelberg (DPO) and Stackelberg (GRPO), we run $T = 8$ iterations with $D = 64$ duels per iteration. For the EXP3 leader we set exploration rate $\gamma = 0.2$. The difficulty reward uses score threshold $\tau_{\text{th}} = 3.0$, and the preference quality reward uses gap schedule $g_{\text{start}} = 0.6$, $g_{\text{end}} = 0.15$, $\sigma_r = 0.15$; the combined reward weights are $\lambda_1 = 0.3$ (difficulty) and $\lambda_2 = 0.7$ (preference quality). For opponent matching we use $\sigma_g = 0.15$ and random override probability $p_{\text{rand}} = 0.2$, with a candidate pool of top-3 by reputation proximity. The reputation system uses $K_0 = 10$, $K_{\min} = 5$, decay rate $\alpha = 0.9$, decay step size $n_s = 10$, scaling factor $\lambda = 20$, and Elo floor $\epsilon = 0.01$. For DPO training, we use learning rate 1×10^{-6}, 1 epoch per iteration, and effective batch size 16 with LoRA. For GRPO training, we use the same learning rate and epoch schedule with $G = 4$ rollouts per prompt; all experiments run on NVIDIA H100 GPUs.

Datasets. We evaluate across 12 datasets in five domains. *Scientific* (4): BixBench (Mitchener et al., 2025), LabBench (Laurent et al., 2024), SMDD (Han et al., 2026), and AssayBench (De Brouwer et al., 2026), spanning biology, bioinformatics, and drug discovery. *Reasoning* (2): GPQA-Diamond (Rein et al., 2024) (graduate-level science) and MATH (Hendrycks et al., 2021). *Code* (2):

Table 1: Performance across three model pools and 12 datasets grouped by domain. **Bold**: best per column within each pool; underline: second best; –: not applicable. Shaded rows ( ) are STACKELBERG ALIGNMENT variants. The Avg column is the macro-average over all available datasets, with AlpacaEval min-max normalized to $[0, 1]$.

	Scientific				Reasoning		Code		Instruction		Knowledge		
Method	*BixBench*	*LabBench*	*SMDD*	*AssayBench*	*GPQA-Dia*	*MATH*	*HumanEval*	*MBPP*	*AlpacaEval*	*IFEval*	*TruthfulQA*	*CulturalBench*	*Avg*
Pool 1: Specialized Expert LLMs													
Sparta Alignment	0.301	0.308	0.168	0.017	0.313	0.820	0.719	0.634	7.528	0.707	**0.686**	0.656	0.524
Majority Vote	0.184	0.272	–	–	0.263	0.778	–	–	–	–	0.616	0.670	0.464
AggLM	0.291	0.317	0.007	0.029	0.273	0.817	0.649	0.565	−2.647	0.469	0.605	0.591	0.407
Trained Router	0.233	0.263	0.011	0.015	0.313	0.735	0.623	0.511	1.718	0.631	0.605	0.633	0.428
Multiagent FT	0.223	0.316	0.187	0.019	0.273	0.729	0.640	0.579	5.785	0.625	0.585	0.682	0.475
Multiagent Debate	0.319	0.292	0.011	0.015	0.353	0.789	0.711	0.608	−6.553	0.567	0.571	0.732	0.414
Het. Swarms	0.194	0.264	0.000	**0.033**	**0.364**	0.789	0.597	0.495	−3.633	0.740	0.622	0.746	0.420
MoA	0.223	0.287	0.095	0.016	0.353	0.833	**0.816**	0.589	−3.848	0.707	0.582	0.729	0.451
Stackelberg (DPO)	0.233	0.311	0.204	0.024	**0.364**	0.857	0.763	0.643	**8.081**	**0.749**	0.682	**0.747**	0.548
Stackelberg (GRPO)	**0.349**	**0.322**	**0.237**	0.023	0.353	**0.877**	**0.816**	**0.653**	7.819	0.730	0.684	0.724	**0.563**
Pool 2: LLMs from Diverse Academic Research													
Sparta Alignment	0.272	0.330	0.216	0.019	0.303	0.853	0.789	0.628	**6.273**	0.672	0.632	0.680	0.533
Majority Vote	0.146	0.287	–	–	0.263	0.825	–	–	–	–	0.650	0.657	0.471
AggLM	0.282	0.282	0.011	0.015	0.283	0.727	0.439	0.382	−1.835	0.416	0.551	0.582	0.341
Trained Router	0.214	0.261	0.170	0.015	0.343	0.826	0.702	0.616	3.066	0.649	0.614	0.654	0.476
Multiagent FT	0.233	0.339	0.156	0.015	0.303	0.837	0.754	0.604	3.564	0.623	0.624	0.683	0.490
Multiagent Debate	0.280	0.311	0.096	0.014	0.313	0.805	0.693	0.622	−2.921	0.583	0.574	0.353	0.387
Het. Swarms	**0.311**	0.303	0.182	**0.028**	0.374	0.884	0.693	0.598	0.440	**0.721**	0.638	0.689	0.482
MoA	0.252	0.310	0.164	0.016	0.414	0.826	0.737	0.616	−0.524	0.681	0.530	0.686	0.458
Stackelberg (DPO)	**0.311**	0.334	0.196	0.026	0.393	**0.889**	**0.816**	0.632	4.986	0.676	**0.656**	**0.696**	0.540
Stackelberg (GRPO)	0.291	**0.355**	**0.234**	0.026	**0.423**	0.884	**0.816**	**0.634**	5.207	0.680	0.626	0.680	**0.544**
Pool 3: General-Purpose LLMs													
Sparta Alignment	**0.349**	0.218	0.294	**0.034**	0.384	0.854	**0.868**	0.661	7.109	0.775	0.770	0.749	0.579
Majority Vote	0.214	0.171	–	–	–	–	–	–	–	–	–	–	0.192
AggLM	0.330	**0.346**	0.120	0.031	0.313	0.857	0.535	0.550	−2.975	0.569	**0.789**	0.790	0.436
Trained Router	0.243	0.173	0.073	0.016	0.292	0.825	0.781	0.606	5.888	0.589	0.681	0.667	0.485
Multiagent FT	0.340	0.263	0.139	0.027	0.208	0.494	0.447	0.413	3.603	0.655	0.682	0.683	0.417
Multiagent Debate	0.307	0.152	0.348	0.027	0.125	0.890	0.614	0.626	−1.965	0.765	0.707	0.830	0.458
Het. Swarms	0.312	0.181	0.265	0.030	0.208	0.853	0.860	**0.691**	−2.623	0.736	0.357	0.803	0.444
MoA	0.291	0.257	0.303	0.024	0.375	**0.906**	0.500	0.626	−1.233	0.769	0.770	**0.853**	0.487
Stackelberg (DPO)	0.322	0.309	**0.390**	0.030	0.394	0.882	0.860	0.663	4.106	0.778	0.770	0.810	0.576
Stackelberg (GRPO)	0.340	0.330	0.353	0.028	**0.424**	0.869	**0.868**	0.684	**7.138**	**0.791**	0.780	0.841	**0.609**

HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). *Instruction following* (2): AlpacaEval (Dubois et al., 2024) (score from a reward model, Skywork-Reward-Llama-3.1-8B-v0.2 (Liu et al., 2024)) and IFEval (Zhou et al., 2023) (verifiable instruction-following accuracy). *Knowledge and truthfulness* (2): TruthfulQA (Lin et al., 2022) and CulturalBench-Hard (Chiu et al., 2024).

4 RESULTS

Table 1 reports performance of all methods across the three model pools and 12 datasets.

Stackelberg consistently outperforms all baselines on average. STACKELBERG ALIGNMENT achieves the highest macro-average across all three pools (Table 1). Stackelberg (GRPO) ranks first with Avg scores of 0.563, 0.544, and 0.609 in Pools 1–3, while Stackelberg (DPO) ranks second in Pools 1 and 2 (0.548 and 0.540). Relative to the strongest training-based baseline, Sparta Alignment, Stackelberg (GRPO) improves macro-average by 7.4%, 2.1%, and 5.2% in the three pools. The gap over the best static inference baseline (MoA or Heterogeneous Swarms) is substantially larger, exceeding 20% in Pools 1 and 3 and 12% in Pool 2, confirming that (1) training-based collaboration methods are generally stronger and (2) STACKELBERG ALIGNMENT further improves upon the state of the art by adaptively concentrating training on informative instructions. Appendix B.1 reports per-task 95% confidence intervals for Stackelberg (GRPO) and (DPO), confirming that many of these individual gains are statistically significant.

Consistent gains on open-ended tasks. The most consistent column-wise advantage appears on SMDD (Small Molecule Drug Discovery), an open-ended scientific task: Stackelberg (GRPO) and (DPO) rank first and second on SMDD across all three pools. On AlpacaEval, Stackelberg achieves the highest reward-model score in Pools 1 and 3 (8.081 and 7.138). IFEval follows similarly, with Stackelberg (DPO) first in Pool 1 and Stackelberg (GRPO) first in Pool 3. A common thread is that these tasks are open-ended and lack a single correct answer, making the quality of the peer-judgment

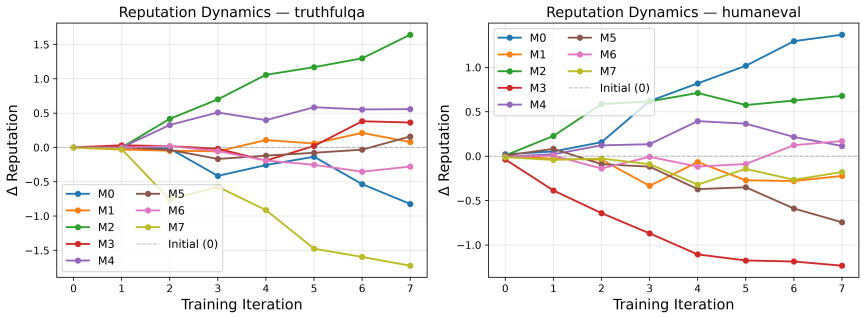


Figure 3: Reputation score trajectories over training iterations for TruthfulQA and HumanEval (Pool 2, Stackelberg GRPO). Reputation scores diverge from the same initialization and stabilize by iteration 4–5. The leading model differs across tasks, confirming genuine task-specific competitive structure and collaborative learning landscape.

signal central to training progress; the EXP3 curriculum concentrates duels on instructions where the model pool still disagrees, producing richer preference pairs exactly where they matter most.

Gains transfer broadly across reasoning, code, and knowledge domains. The improvement extends well beyond the tasks most directly linked to preference learning. On GPQA-Diamond, Stackelberg (GRPO) leads in Pools 2 and 3 (0.423 and 0.424) and Stackelberg (DPO) ties for first in Pool 1 alongside Heterogeneous Swarms. On MATH, Stackelberg (GRPO) achieves the best score in Pool 1 (0.877) and Stackelberg (DPO) leads in Pool 2 (0.889); MATH instructions span a wide difficulty range, and we hypothesize this is exactly the setting where an adaptive curriculum that concentrates duels at the frontier of the pool's current ability (§2.1) has the most room to help relative to fixed uniform sampling. On HumanEval, Stackelberg (GRPO) ties for first in all three pools. Exceptions occur on narrow scientific tasks: LabBench in Pool 3 is led by AggLM, and AssayBench in Pools 1–2 is led by Heterogeneous Swarms. This suggests that highly specialized benchmarks like scientific discovery may benefit from more high-quality data in the instruction pool.

GRPO and DPO show complementary strengths. Within our two training variants, Stackelberg (GRPO) outperforms Stackelberg (DPO) on Avg in all pools and on most scientific and reasoning columns, while Stackelberg (DPO) is more competitive on instruction-following and cultural knowledge tasks (IFEval and CulturalBench in Pool 1; TruthfulQA and CulturalBench in Pool 2). Notably, Stackelberg (DPO) falls slightly below Sparta in Pool 3 (0.576 vs. 0.579), the one exception to Stackelberg's consistent advantage, suggesting that the GRPO objective provides a signal that DPO alone does not reliably deliver in this setting.

Margins vary with pool heterogeneity. Among non-Stackelberg training baselines, Sparta Alignment is consistently the strongest, validating the duel-based preference learning framework; the additional gains from the adaptive leader over Sparta range from 2.1% to 7.4%. The narrowest margin appears in Pool 2, where the heterogeneous academic training backgrounds of the models make it harder to find a single instruction distribution that benefits all models simultaneously. Pool 1 (specialized expert LLMs) shows the largest GRPO margin over Sparta (7.4%), while Pool 3 (four general-purpose LLMs) shows the largest gap over static inference (25.1% above MoA).

5 Analysis

Pool heterogeneity and competitive stratification. To achieve competitive pool training in Stackelberg Alignment, models must differ by capabilities. Figure 2 (Pool 2, GRPO) confirms this: no model dominates all tasks, and the specialist structure is clear: `M4` leads on 5/10 tasks but loses to `M0` on reasoning and coding and to `M1` on TruthfulQA. This heterogeneity is captured by the reputation system: Figure 3 shows reputation trajectories for TruthfulQA and HumanEval. Reputation scores gradually diverge from equal starting values, stabilize by iteration 4–5, and the winning model differs across tasks, providing evidence that task-specific competitive structure is resolved early and maintained throughout training.

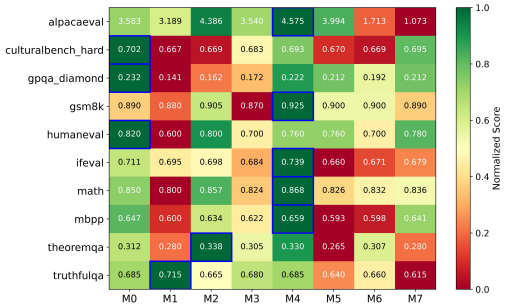


Figure 2: Per-model eval scores at best iteration (Pool 2, Stackelberg GRPO). Row-normalized; green = best, red = worst per task.

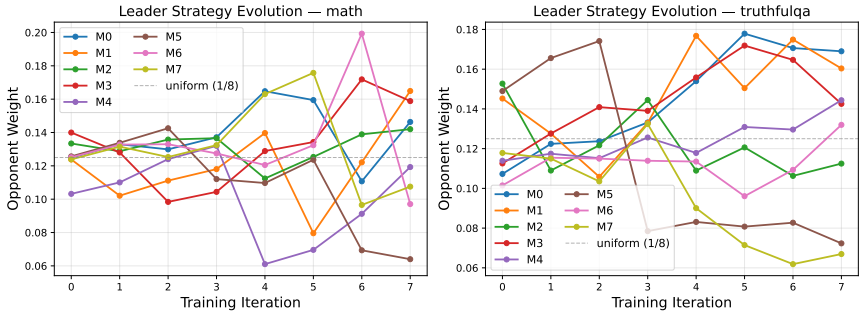


Figure 4: Normalized opponent weight assigned by the Stackelberg leader over iterations (Pool 2, Stackelberg GRPO). Dashed line = uniform selection baseline. The leader shields the emerging dominant model from being excessively used as an opponent and adapts its strategy per task.

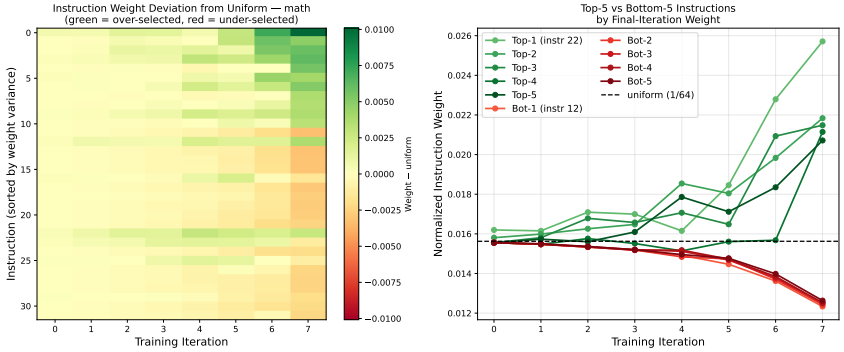


Figure 5: Leader instruction weights over training iterations (the MATH dataset, model pool 1). Left: heatmap of normalized weight deviation from uniform for the top-32 most dynamic instructions. Right: trajectory of top-5 and bottom-5 instructions by final-iteration weight; dashed line = uniform. The leader converges to an implicit difficulty curriculum by iteration 5.

Adaptive opponent selection. Given this heterogeneity, the competitive landscape the leader must navigate is non-trivial: pairwise win rates on MATH span 0.1–0.6 with no model dominating (Appendix B.3, Figure 6), and the leader exploits this structure through adaptive opponent selection. Figure 4 traces the leader's normalized opponent weights over iterations for math and TruthfulQA. Early choices are already task-specific: math initially concentrates on `M3`; TruthfulQA peaks on `M5` at iteration 2. Notably, the model with the highest final eval score is actively shielded as an opponent at peak training time, with its weight dropping to $< 0.5\times$ uniform at iterations 4–5, consistent with the Stackelberg objective of not wasting training budget on unwinnable duels. The strategy is genuinely task-adaptive: the same model (`M7`) is the most-selected opponent on math (weight 0.176 at iteration 5) but nearly irrelevant on TruthfulQA (weight 0.062 at iteration 7).

Emergent difficulty curriculum over instructions. Figure 5 shows the leader's per-instruction weights over training iterations (math task, Pool 1). Iterations 0–4 are near-uniform: the leader explores all instructions before committing. By iteration 5, the distribution sharpens: the top-5 instructions rise to $\sim$1.6$\times$ uniform by iteration 7, while the bottom-5 fall to $\sim$0.75$\times$ uniform, and the divergence exists and is accelerating. This constitutes an emergent difficulty curriculum: the leader identifies instructions where model disagreement persists, precisely where preference signal is most informative, and concentrates duels there without any explicit difficulty label. The behavior mirrors the leader's opponent-selection strategy (Fig. 4): both start near-uniform and progressively concentrate attention as the competitive hierarchy becomes clearer.

6 Conclusion

We presented Stackelberg Alignment, a collaborative alignment framework for adaptive instruction selection and curriculum learning in multi-LLM evolution. By casting the training loop as a Stackelberg game solved via EXP3, Stackelberg Alignment concentrates pairwise duels on the instructions where the model pool still disagrees. Across three heterogeneous model pools and 12 benchmarks, Stackelberg Alignment with GRPO learning achieves the highest macro-average in every pool, improving over training-based baselines by up to 7.4% and over the best static inference baseline by 12–25%, with ablations confirming that the adaptive curriculum, reputation-weighted judgment, and reputation-based matching each contribute to these gains.

AI USE STATEMENT

We used generative AI tools to help with language polishing. The AI tools were used solely for language editing and did not contribute to the intellectual content of the work. We have reviewed all AI-assisted edits and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

All hyperparameters are reported in Appendix C (Implementation Details), including EXP3 learning rate and reward coefficients, LoRA configuration, DPO and GRPO training settings, and generation parameters. Dataset sizes and splits are provided in Appendix E (Dataset Statistics). Baseline configurations are listed in Appendix F (Baseline Configurations), and the judge prompt is presented in Appendix G (Judge Prompt). Statistical significance for our main results is reported in Appendix B.1. Our method builds on publicly available pretrained models and standard training libraries; no proprietary data or infrastructure is required beyond the compute described in Appendix C.

ETHICS STATEMENT

STACKELBERG ALIGNMENT is designed to improve language model alignment through collaborative training: models improve each other via structured competition and peer judgment, which is broadly beneficial. However, competitive training pools also introduce potential misuse scenarios. A malicious actor who controls one model in the pool could manipulate duel outcomes to inflate its own reputation score, skew peer judgment, and inject adversarial preference pairs into the training data, degrading the alignment of all other models in the pool (Yang et al., 2026). The reputation system provides partial robustness by down-weighting low-reputation judges, but does not make it a formal defense against adversarial participants. We see adversarial robustness of collaborative training as an important open problem, particularly as decentralized or federated model collaboration systems are deployed at scale.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1998.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, et al. Nvidia nemotron 3: Efficient and open intelligence. *arXiv preprint arXiv:2512.20856*, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, et al. CulturalBench: A robust, diverse, and challenging cultural benchmark by human-ai culturalteaming. *arXiv preprint arXiv:2410.02677*, 2024.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*, 2024.

Edward De Brouwer, Carl Edwards, Alexander Wu, Jenna Collier, Graham Heimberg, Xiner Li, Meena Subramaniam, Ehsan Hajiramezanali, David Richmond, Jan-Christian Hütter, et al. AssayBench: An assay-level virtual cell benchmark for llms and agents. *arXiv preprint arXiv:2605.10876*, 2026.

Yilun Du and Leslie Pack Kaelbling. Position: Compositional generative modeling: A single model is not all you need. In *Forty-first International Conference on Machine Learning*, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Aram Ebtekar and Paul Liu. Elo-mmr: A rating system for massive multiplayer competitions. In *Proceedings of the Web Conference 2021*, pp. 1772–1784, 2021.

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

Shangbin Feng, Zifeng Wang, Palash Goyal, Yike Wang, Weijia Shi, Huang Xia, Hamid Palangi, Luke Zettlemoyer, Yulia Tsvetkov, Chen-Yu Lee, et al. Heterogeneous swarms: Jointly optimizing model roles and weights for multi-llm systems. *arXiv preprint arXiv:2502.04510*, 2025a.

Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Shannon Zejiang Shen, Xiaochuang Han, Hunter Lang, Chen-Yu Lee, et al. When one LLM drools, multi-LLM collaboration rules. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17048–17063, 2026a.

Shangbin Feng, Yike Wang, Weijia Shi, Luke Zettlemoyer, Yejin Choi, and Yulia Tsvetkov. Scaling participation in modular ai systems. *arXiv preprint arXiv:2606.07812*, 2026b.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Kevin Han, Renfei Zhang, Kathy Wei, Hamed Mahdavi, Niloofar Mireshghallah, and Amir Barati Farimani. SMDD-Bench: Can llms solve real-world small molecule drug design tasks? *arXiv preprint arXiv:2605.21740*, 2026.

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. In *The Thirty-fifth Annual Conference on Neural Information Processing Systems*, 2021.

Yuru Jiang, Wenxuan Ding, Shangbin Feng, Greg Durrett, and Yulia Tsvetkov. Sparta alignment: Collectively aligning multiple language models through combat. *Advances in Neural Information Processing Systems*, 38:132970–133009, 2025.

Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. LAB-Bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Y Zou, and Pan Lu. In-the-flow agentic system optimization for effective planning and tool use. In *International Conference on Learning Representations*, volume 2026, pp. 50524–50570, 2026.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Bo Liu, Simon Yu, Yiding Jiang, Ao Qu, Andrew Zhao, Zichen Liu, Junsu Kim, Zijian Zhou, Seungone Kim, Tongzheng Ren, et al. Spade: Self-play in adaptive synthetic executable environments. *arXiv preprint arXiv:2608.19197*, 2026a.

Bo Liu, Simon Yu, Zichen Liu, Leon Guertler, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 35407–35434, 2026b.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Qingwei Lin, Jianguang Lou, Shifeng Chen, Yansong Tang, and Weizhu Chen. Arena learning: Build data flywheel for llms post-training via simulated chatbot arena. *arXiv preprint arXiv:2407.10627*, 2024.

Ludovico Mitchener, Jon M Laurent, Alex Andonian, Benjamin Tenmann, Siddharth Narayanan, Geemi P Wellawatte, Andrew White, Lorenzo Sani, and Samuel G Rodriques. BixBench: A comprehensive benchmark for llm-based agents in computational biology. *arXiv preprint arXiv:2503.00096*, 2025.

Remi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 36743–36768, 2024.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

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

Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707*, 2025.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Vinzenz Thoma, Barna Pásztor, Andreas Krause, Giorgia Ramponi, and Yifan Hu. Contextual bilevel reinforcement learning for incentive alignment. *Advances in Neural Information Processing Systems*, 37:127369–127435, 2024.

H.F. von Stackelberg and A.T. Peacock. *The Theory of the Market Economy*. Hodge, 1952.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025a.

Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. Cream: Consistency regularized self-rewarding language models, 2025b.

Chu Xu, Zhixin Zhang, Tianyu Jia, and Yujie Jin. Stackelberg self-annotation: A robust approach to data-efficient llm alignment. *Advances in Neural Information Processing Systems*, 38:62912–62949, 2025.

Ziyuan Yang, Wenxuan Ding, Shangbin Feng, and Yulia Tsvetkov. Among us: Measuring and mitigating malicious contributions in model collaboration systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15969–15988, 2026.

Jihan Yao, Wenxuan Ding, Shangbin Feng, Lucy Lu Wang, and Yulia Tsvetkov. Varying shades of wrong: Aligning llms with wrong answers only, 2024.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2025.

Xiang Yue, Yueqi Song, Akari Asai, Seungone Kim, JEAN NYANDWI, Simran Khanuja, Anjali Kantharuban, Lintang Sutawika, Sathyanarayanan Ramamoorthy, and Graham Neubig. Pangea: A fully open multilingual multimodal llm for 39 languages. In *International Conference on Learning Representations*, volume 2025, pp. 47758–47811, 2025.

Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. Cures: From gradient analysis to efficient curriculum learning for reasoning llms. In *International Conference on Learning Representations*, volume 2026, pp. 1531–1556, 2026.

Yuheng Zhang, Dian Yu, Baolin Peng, Linfeng Song, Ye Tian, Mingyue Huo, Nan Jiang, Haitao Mi, and Dong Yu. Iterative nash policy optimization: Aligning llms with general preferences via no-regret learning. In *International Conference on Learning Representations*, volume 2025, pp. 31833–31849, 2025.

Wenting Zhao, Pranjal Aggarwal, Swarnadeep Saha, Asli Celikyilmaz, Jason Weston, and Ilia Kulikov. The majority is not always right: Rl training for solution aggregation. *arXiv preprint arXiv:2509.06870*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Related Work

Model collaboration. The landscape of multi-LLM collaboration can be organized by the level of information exchange (Feng et al., 2026b). API-level methods route or cascade queries to the most suitable model in the pool (Ong et al., 2025; Feng et al., 2025b; Chen et al., 2023). Text-level methods have models exchange generated text: through debate and multi-round discussion (Du et al., 2024), response synthesis across model layers (Wang et al., 2025a), structured interaction graphs (Feng et al., 2025a), and modular value alignment (Feng et al., 2024). Weight-level methods operate directly in parameter space via model merging. Among training-based approaches, Multiagent Fine-tuning (Subramaniam et al., 2025) uses debate-generated data for supervised fine-tuning; AggLM (Zhao et al., 2025) trains an RL aggregator over model outputs; Arena Learning (Luo et al., 2024) simulates pairwise ELO-rated battles to generate post-training data; and Sparta Alignment (Jiang et al., 2025) trains models through competitive duels with peer judgment. Stackelberg Alignment extends this line by introducing an adaptive leader that determines *which instructions* drive collaborative training at each step, turning the instruction distribution itself into a learned, non-stationary curriculum.

Self-play and curriculum learning. A related line of work trains LLMs through self-play and iterative self-improvement. SPIN (Chen et al., 2024) and Self-Rewarding LMs (Yuan et al., 2025) improve a single model by having it compete against or judge its own prior outputs. SPIRAL (Liu et al., 2026b) extends self-play to zero-sum multi-agent games, generating an automatic opponent curriculum; SPADE (Liu et al., 2026a) trains a single model as both environment designer and reasoning agent, adaptively targeting its own capability frontier. A parallel thread develops curriculum learning for LLM fine-tuning: easy-to-hard generalization (Sun et al., 2024) shows that ordering training examples by difficulty improves reasoning; SEC (Chen et al., 2025) frames curriculum selection as a non-stationary bandit over instruction categories; CurES (Zeng et al., 2026) uses gradient signals to identify the most informative training examples at each stage. Stackelberg Alignment unifies these threads: it applies bandit-driven instruction selection within a competitive multi-LLM training loop, where the difficulty signal emerges from inter-model disagreement rather than from predefined categories or gradient heuristics.

Game-theoretic alignment. Our work is grounded in game-theoretic alignment, where the interaction between alignment objectives and model responses is formalized as a strategic game between rational agents rather than a fixed optimization target. Nash Learning from Human Feedback (Munos et al., 2024) formalizes preference optimization as a two-player constant-sum game with the Nash policy as solution; INPO (Zhang et al., 2025) iteratively approximates the Nash policy via no-regret online learning. These Nash-based methods model alignment as a symmetric simultaneous game. Stackelberg Alignment instead adopts a Stackelberg formulation (von Stackelberg & Peacock, 1952; Başar & Olsder, 1998): the leader (instruction selector) commits to a distribution before the followers (LLMs) respond and train on such data, capturing the natural asymmetry between curriculum design and model adaptation. SSAPO (Xu et al., 2025) also applies Stackelberg games to LLM alignment, but as a two-policy robustness problem against noisy preference labels; Stackelberg Alignment applies the framing to a multi-follower competitive pool where the leader maximizes instructional informativeness. Bilevel RL for incentive alignment (Thoma et al., 2024) provides further theoretical grounding for leader-follower RL under best-response constraints.

B Additional Results

B.1 Statistical Significance

Table 2 reports per-task 95% confidence intervals for Stackelberg (GRPO) and (DPO) across all three pools, computed via 10,000 Monte Carlo draws (Bernoulli for binary tasks; Gaussian for AlpacaEval with $\sigma = 5.0/\sqrt{n}$). A * mark indicates the point estimate exceeds every baseline on that task; ** indicates the CI lower bound does. Stackelberg (GRPO) earns at least one ** mark in every pool, and at least one * mark on 7–8 of the 10 reported tasks. SMDD and AssayBench are excluded: their small test sets ($n \leq 334$) inflate Gaussian CIs to ± 0.55, making interval comparisons uninformative.

Table 2: Per-task scores and 95% confidence intervals for Stackelberg (GRPO) and Stackelberg (DPO). CIs computed via 10,000 Monte Carlo draws (Bernoulli for binary tasks; Gaussian for AlpacaEval). *: point estimate exceeds every baseline. **: CI lower bound exceeds every baseline. SMDD and AssayBench omitted (CI $\approx \pm 0.55$).

	Stackelberg (GRPO)		Stackelberg (DPO)	
Task	Score	95% CI	Score	95% CI
Pool 1: Specialized Expert LLMs				
BixBench	0.350^{*}	[0.262, 0.447]	0.233	[0.155, 0.320]
LabBench	0.322^{*}	[0.284, 0.360]	0.311	[0.273, 0.348]
GPQA-Dia	0.354	[0.263, 0.444]	0.364	[0.273, 0.455]
MATH	0.877^{**}	[0.856, 0.898]	0.857^{**}	[0.834, 0.879]
HumanEval	0.816	[0.746, 0.886]	0.763	[0.684, 0.842]
MBPP	0.653^{*}	[0.610, 0.696]	0.643	[0.602, 0.684]
AlpacaEval	7.819^{*}	[7.512, 8.127]	8.081^{**}	[7.771, 8.397]
IFEval	0.730	[0.676, 0.784]	0.749^{*}	[0.692, 0.800]
TruthfulQA	0.684	[0.647, 0.720]	0.682	[0.645, 0.718]
CulturalBench	0.724	[0.687, 0.759]	0.747^{*}	[0.711, 0.781]
Pool 2: Diverse Academic Research				
BixBench	0.291	[0.204, 0.379]	0.311	[0.223, 0.398]
LabBench	0.355^{*}	[0.317, 0.395]	0.334	[0.296, 0.374]
GPQA-Dia	0.423^{*}	[0.323, 0.515]	0.393	[0.293, 0.495]
MATH	0.884	[0.863, 0.904]	0.889^{*}	[0.868, 0.909]
HumanEval	0.816^{*}	[0.746, 0.886]	0.816^{*}	[0.746, 0.886]
MBPP	0.634	[0.591, 0.678]	0.632	[0.589, 0.674]
AlpacaEval	5.207	[4.897, 5.514]	4.986	[4.676, 5.303]
IFEval	0.680	[0.620, 0.736]	0.676	[0.620, 0.732]
TruthfulQA	0.626	[0.588, 0.663]	0.656	[0.618, 0.692]
CulturalBench	0.680	[0.643, 0.718]	0.696^{*}	[0.659, 0.731]
Pool 3: General-Purpose LLMs				
BixBench	0.340	[0.252, 0.437]	0.322	[0.233, 0.418]
LabBench	0.330	[0.291, 0.370]	0.309	[0.272, 0.346]
GPQA-Dia	0.424	[0.323, 0.525]	0.394	[0.303, 0.495]
MATH	0.869	[0.848, 0.890]	0.882	[0.861, 0.902]
HumanEval	0.868	[0.807, 0.930]	0.860	[0.790, 0.921]
MBPP	0.684	[0.641, 0.725]	0.663	[0.620, 0.706]
AlpacaEval	7.138	[6.830, 7.449]	4.106	[3.796, 4.417]
IFEval	0.791^{*}	[0.740, 0.840]	0.778^{*}	[0.724, 0.828]
TruthfulQA	0.780	[0.746, 0.812]	0.770	[0.736, 0.802]
CulturalBench	0.841	[0.811, 0.870]	0.810	[0.778, 0.842]

B.2 Cross-Task Transfer

Table 3 tests whether competitive pool training on one task transfers to others. We take the best Stackelberg (GRPO) checkpoint trained exclusively on MATH-type instructions (Pool 2, `parti_4_full`, iteration 6) and evaluate it on HumanEval and MBPP; symmetrically, we take the best checkpoint trained on HumanEval instructions (`parti_0_full`, iteration 7) and evaluate it on MATH and MBPP. The baseline is the best untrained single model in the pool per task.

Training on MATH transfers dramatically to coding (+18 pp on HumanEval, +33 pp on MBPP), while same-task pool training on coding yields no improvement over the best untrained model. HumanEval pool training likewise transfers to MBPP (+32 pp). These results suggest that competitive training on mathematical reasoning produces broadly transferable gains rather than task-specific memorization.

Table 3: Cross-task transfer in Pool 2. Each trained column shows the best single-model checkpoint from competitive pool training on that source task, evaluated on the target. Baseline: best untrained model in the pool per task. "No gain" indicates same-task training matched or fell below the untrained baseline.

		Checkpoint source		
Eval Task	Best Untrained	Same-Task	Math-Trained	HumanEval-Trained
MATH	0.863	0.865	–	0.866
HumanEval	0.790	0.790 (no gain)	**0.974** (+18pp)	–
MBPP	0.639	0.632 (no gain)	**0.963** (+33pp)	**0.961** (+32pp)

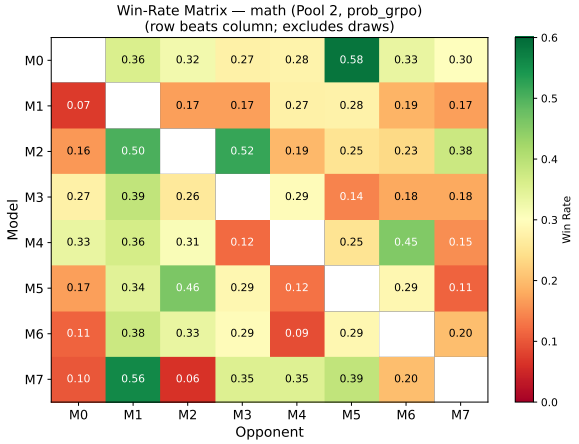


Figure 6: Pairwise win-rate matrix for the MATH dataset (Pool 2, Stackelberg GRPO). No model dominates; win rates span 0.1–0.6; ∼41% of duels are draws given the objective nature of the task.

B.3 Ablation: Opponent Selection Strategy

Figure 6 shows the pairwise win-rate matrix for MATH (Pool 2, Stackelberg GRPO) referenced in §5: no model dominates, win rates span 0.1–0.6, and ∼41% of duels are draws given the objective nature of the task, motivating the adaptive opponent selection studied below.

Figure 7 compares four opponent-selection strategies on the MATH task (Pool 2). All four beat the single-model baseline (0.857), confirming that competitive training helps regardless of opponent schedule. `schedule_increasing` (0.878) outperforms the default `schedule_decreasing` (0.865), suggesting that gradually concentrating on harder opponents is beneficial. `lowest_diff` (0.875) also outperforms the default, while `highest_diff` (0.861) performs worst; overexposure to very hard opponents appears noisy. Note that the three ablation variants ran for 2 iterations vs. 8 for the default; their final scores are not directly comparable but are reported for reference.

B.4 Ablation: Pool Size

Figure 8 compares pool sizes of 4, 6, and 8 models on the MATH task. All three beat the single-model baseline. Pool 6 (0.880) slightly outperforms Pool 4 (0.875) and Pool 8 (0.865), suggesting a moderate pool size is preferable; however, CIs overlap between Pool 4 and Pool 6. Larger pools do not monotonically help; Pool 8 underperforms both smaller configurations, consistent with the intuition that very large pools dilute the competitive signal each model receives.

B.5 Response Diversity During Training

Figure 9 shows lexical diversity (type-token ratio, TTR) and mean response length per model over training iterations (MATH task, Pool 2). TTR oscillates stably across all 8 iterations with no downward trend, and response length remains in the 100–135 word range throughout. Between-model

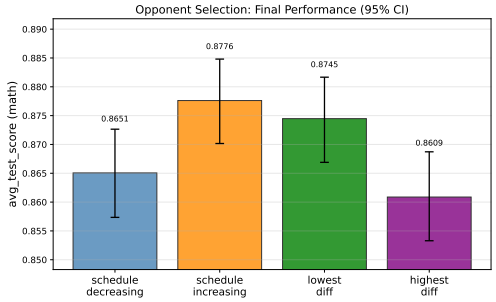


Figure 7: Opponent selection ablation on MATH (Pool 2). All strategies beat the single-model baseline (0.857); `schedule_increasing` is best.

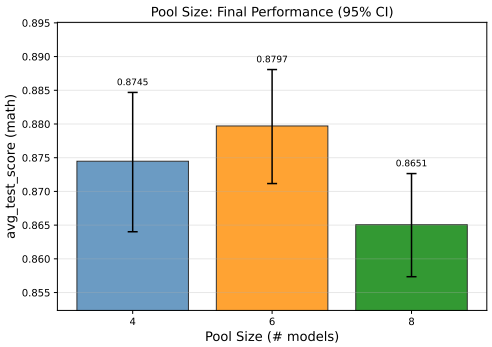


Figure 8: Pool size ablation on MATH (Pool 2). All sizes outperform the single-model baseline (0.857); Pool 6 is best.

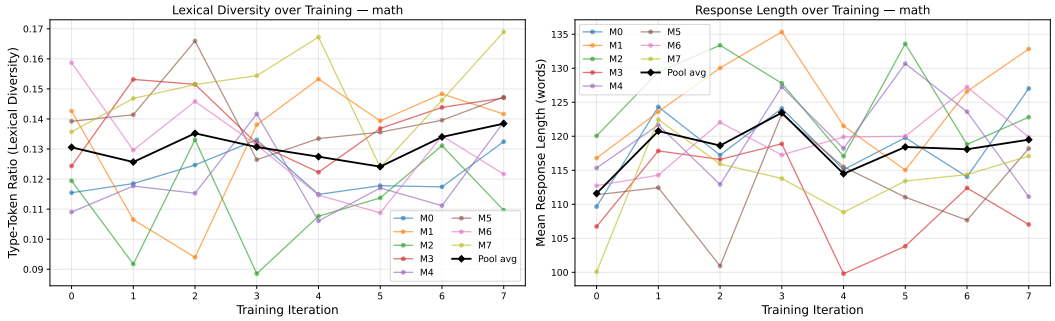


Figure 9: Response properties over training iterations (MATH, Pool 2, 8 models). Left: type-token ratio (lexical diversity). Right: mean response length. Bold line = pool average. No model collapses toward repetitive or length-degenerate outputs; pool diversity is maintained throughout.

variation dominates within-model trends, indicating that stylistic properties of the pre-trained models persist despite competitive training. This confirms that the pool maintains behavioral heterogeneity throughout training, a prerequisite for the leader's adaptive selection strategy to remain meaningful.

B.6 Ablation: LLM Leader vs. Probabilistic Leader

Table 4 compares two leader strategies on Pool 1 trained with DPO and GRPO across four shared tasks. The *LLM leader* uses Gemini to select instructions based on model embeddings and current reputation scores (pilot experiment on the same Pool 1 models). The *probabilistic leader* uses EXP3, an adversarial bandit algorithm that maintains a sampling distribution over instructions updated by follower win-rates; results are taken from Table 1.

Table 4: Leader strategy comparison on Pool 1 across four tasks. Best score per task is **bolded**. AlpacaEval is scored by a reward model (higher = better); all other tasks report accuracy.

Method	AlpacaEval	GPQA	MBPP	TruthfulQA
Gemini Leader + DPO	1.65	0.263	0.581	0.682
Prob. Leader + DPO	**8.081**	**0.364**	0.643	0.682
Prob. Leader + GRPO	7.819	0.353	**0.653**	**0.684**

The probabilistic leader substantially outperforms the Gemini leader on AlpacaEval (8.08 vs. 1.65) and GPQA (+10.1% absolute), with consistent gains on MBPP (+6.2% for DPO, +7.2% for GRPO) and essentially matched performance on TruthfulQA. These results suggest that the expressivity of an LLM leader does not translate to better instruction selection in practice: the EXP3 bandit, which adapts directly to observed win-rates without requiring generalization from model embeddings, yields more effective curricula. The probabilistic leader also avoids the latency and cost of LLM inference at each training round. Between the two probabilistic conditions, GRPO edges out DPO on MBPP and TruthfulQA while DPO leads on GPQA, consistent with the complementary strengths in Table 1.

C Implementation Details

EXP3 leader. We run $T{=}8$ training iterations. The leader maintains a weight vector $\mathbf{w}^t \in \mathbb{R}^K_{>0}$ over an instruction pool of size $K{=}64$. The sampling distribution mixes exploitation and exploration with rate $\gamma{=}0.2$ (Eq. 1). The composite reward is $r_k = 0.3\, r_{\text{diff}} + 0.7\, r_{\text{pref}}$, where r_{diff} is the difficulty reward (Eq. 2) with floor threshold $\tau_{\text{th}}{=}3.0$ (on the 1–10 judge scale), and r_{pref} is the preference-quality reward (Eq. 4) with Gaussian bandwidth $\sigma_r{=}0.15$ on the normalized score gap. The ideal gap target is annealed from 0.6 to 0.15 across training. All scores are normalized to $[0, 1]$ by dividing by $s_{\max} - s_{\min} = 9$.

Reputation system. The K factor decays exponentially with the total number of reputation updates n for each model:

$$K_t = \max\big(K_{\min},\ K_0 \cdot \alpha^{\lfloor n/n_s \rfloor}\big), \tag{10}$$

with $K_0{=}10$, $K_{\min}{=}5$, decay rate $\alpha{=}0.9$, and decay step $n_s{=}10$. This cools the magnitude of reputation swings as models accumulate more duels and their standings stabilize. The reputation deviation σ_i is the sample standard deviation of the last $W{=}20$ reputation deltas:

$$\sigma_i = \sqrt{\frac{1}{|W_i| - 1} \sum_{\Delta \in W_i} (\Delta - \bar{\Delta})^2}, \tag{11}$$

where W_i is the sliding window of the most recent $\min(n, 20)$ updates for model M_i and $\bar{\Delta}$ is their mean. σ_i measures per-model rating volatility and enters the opponent-matching score via z_i (Eq. 9).

Opponent matching. We use `schedule_decreasing` matching: the cosine-scheduled target reputation gap decreases from wide (random-like) to tight (near-peer) as training progresses. A fraction $p_{\text{rand}}{=}0.2$ of duels use uniformly random matching to maintain diversity; each combatant draws from $n_{\text{opp}}{=}3$ candidate opponents.

DPO training. Each model is fine-tuned with LoRA ($r{=}64$, $\alpha{=}16$, dropout 0.1) applied to the query, key, value, and output projections. Optimizer: AdamW, learning rate 10^{-6}, cosine schedule, effective batch size 16 (per-device batch 1, gradient accumulation 16), 1 epoch per iteration. $\beta{=}0.1$, max sequence length 2048, max prompt length 1536.

GRPO training. Same LoRA configuration as DPO. Learning rate 10^{-6}, cosine schedule, per-device batch size 8, 1 epoch, 4 rollout generations per prompt, max completion length 256 tokens. KL coefficient $\beta{=}0.0$; clip ratio $\epsilon{=}0.2$. Judge rewards are normalized to $[0, 1]$ via min-max scaling before being passed to the GRPO objective.

Generation and evaluation. All models generate responses with temperature 0.7, top-p 0.9. Peer judges generate scores greedily (temperature 10^{-5}, max 64 tokens). Evaluation uses task-specific formats: multiple-choice benchmarks are scored by exact match; AlpacaEval uses a reward model; open-ended scientific tasks (SMDD, AssayBench) use domain-specific metrics as described in Section 3. All baselines use identical generation and evaluation settings.

D Model Pool Configurations

Pool 1 and Pool 3 model identities are described in Section 3. Table 5 lists the 8 models comprising Pool 2 (LLMs from Diverse Academic Research), sourced from Feng et al. (2026b).

Table 5: Pool 2 model configurations (LLMs from Diverse Academic Research).

ID	HuggingFace identifier	Specialty
M0	`chtmp223/Qwen2.5-7B-CLIPPER`	Reasoning/alignment
M1	`chengq9/ToolRL-Qwen2.5-3B`	Tool use / RL
M2	`AgentFlow/agentflowplanner-7b`	Agent planning
M3	`nanami/ladder-last16L-llama3.1-8b-instruct-sft4k`	Instruction following
M4	`viswavi/qwen2.5-rlcf`	RL from feedback
M5	`milli19/promptmii-llama3.1-8b-instruct`	Prompt optimization
M6	`Zhengping/conditionalprobability-regression`	Calibration
M7	`yale-nlp/MDCure-Qwen2-7B-Instruct`	Medical

E Dataset Statistics

Table 6 summarizes the instruction pool and test split sizes for each benchmark. The instruction pool (Dev) is drawn from the development set of each benchmark; 80% is used as the pool of candidate instructions available to the EXP3 leader and DPO/GRPO training, and the remaining 20% serves as a held-out validation set for checkpoint selection. The test set is used exclusively for final evaluation.

Table 6: Dataset sizes. Dev: full development set; 80% forms the instruction pool for training and 20% is held out for validation. Test: held-out evaluation set.

Benchmark	Dev	Test
BixBench	102	103
LabBench	578	578
SMDD	135	137
AssayBench	218	334
GPQA-Diamond	99	99
MATH	956	956
HumanEval	50	114
MBPP	487	487
AlpacaEval	2000	1000
IFEval	250	250
TruthfulQA	200	617
CulturalBench	613	613

F Baseline Configurations

All baselines use the same model pools, generation hyperparameters (temperature 0.7, top-p 0.9), and evaluation protocols as Stackelberg Alignment. Training-based baselines (Sparta Alignment, Multiagent FT, AggLM, Trained Router) are given equal gradient updates to control for compute.

Sparta Alignment uses the same duel-based preference learning loop as Stackelberg Alignment but samples instructions uniformly from the same instruction pool, with the same opponent matching schedule and reputation system. **Multiagent FT** collects debate-style responses from all models and fine-tunes each with supervised cross-model distillation using LoRA (lr 10^{-5}, 3 epochs, same LoRA config as our DPO). **AggLM** trains an RL aggregator to select among model outputs; rollout batch size 4. **Trained Router** learns to route each query to the best model via supervised training (1 epoch, cross-entropy on dev-set oracle labels). **Mixture of Agents (MoA)** performs one round of response synthesis (proposer $\rightarrow$ aggregator), inference only. **Heterogeneous Swarms** runs 3 rounds of iterative refinement across the pool with population size 3. **Multiagent Debate** uses 1 round of cross-model critique and revision. **Majority Vote** uses plurality voting across all models at inference.

G JUDGE PROMPT

All peer judges receive the following prompt, filled with the instruction and the response being evaluated. Scores are parsed from JSON output and fall back to regex extraction if needed; the default score on parse failure is 5.0.

```
Please judge the following response based on the question
and the response to be evaluated.
Question: {instruction}
Response to be evaluated: {response}

Operation: Output ONLY a JSON object with one score in this
exact format. Score must be in the range of 1 to 10.
Your output should be like this:
{"score": score}
```

The same prompt is used for both offline DPO judging and online GRPO reward generation. Judge scores are aggregated as a reputation-weighted mean: $\bar{s} = \sum_k R_k s_k / \sum_k R_k$, where R_k is the current reputation score of judge M_k.

H COMPUTATIONAL COMPLEXITY ANALYSIS

Table 7 summarizes the per-iteration theoretical complexity of STACKELBERG ALIGNMENT and key baselines in terms of model forward passes F, training steps S, pool size m, duels per iteration D, instruction pool size K, and dataset size N.

Table 7: Per-iteration theoretical complexity. m: pool size; D: duels per iteration; N: dataset size; K: instruction pool size. "Inference only" methods have $S = 0$.

Method	Forward passes	Training steps
STACKELBERG ALIGNMENT	$O(D \cdot m)$	$O(m)$
Sparta Alignment	$O(D \cdot m)$	$O(m)$
Multiagent FT	$O(N \cdot m)$	$O(m)$
AggLM	$O(N \cdot m)$	$O(1)$
Trained Router	$O(N \cdot m)$	$O(1)$
Mixture of Agents (MoA)	$O(N \cdot m)$	0
Heterogeneous Swarms	$O(N \cdot m \cdot r)$	0
Multiagent Debate	$O(N \cdot m \cdot r)$	0
Majority Vote	$O(N \cdot m)$	0

Discussion. STACKELBERG ALIGNMENT and Sparta Alignment share the same asymptotic complexity: each of the D duels requires two generation passes (the combatants) plus $m-2$ judge passes, giving $O(D \cdot m)$ forward passes per iteration, followed by m independent LoRA training steps. The EXP3 leader adds only an $O(K)$ weight update per duel, which is negligible compared to model forward passes (K=64 instructions vs. billions of parameters per model pass). STACKELBERG ALIGNMENT therefore introduces *zero additional model calls* relative to Sparta Alignment; all gains come from reallocating the fixed duel budget toward more informative instructions, not extra compute.

Multiagent FT, AggLM, and Trained Router each require $O(N \cdot m)$ forward passes because they process all N training instructions with all m models before each training update; with $N \gg D$ (e.g., N=956 vs. D=64 for MATH), this is substantially more expensive per iteration. Inference-only methods (MoA, Heterogeneous Swarms, Debate, Majority Vote) pay no training cost but incur repeated forward passes at inference time; Swarms and Debate multiply by the number of refinement rounds r. In our setting (D=64, $m \leq 9$), the per-iteration forward-pass count for STACKELBERG ALIGNMENT is at most $64 \times 9 = 576$ model calls, vs. up to $956 \times 9 = 8{,}604$ for full-dataset sweeps.

I LIMITATIONS

STACKELBERG ALIGNMENT requires running all models in the pool simultaneously during the combat and judgment phases, which increases memory and compute requirements relative to single-model fine-tuning; this is a shared characteristic of all training-time multi-LLM collaboration methods. The quality and diversity of the instruction pool $\mathcal{X}$ are important: a pool that does not span the capability frontier of the model pool limits the learning signal the EXP3 leader can identify. As models improve across iterations, productive disagreement naturally narrows, suggesting that combining STACKELBERG ALIGNMENT with mechanisms for dynamic instruction augmentation or pool expansion is a promising direction for future work. Finally, the reputation system initializes all models equally and may take several iterations to reliably differentiate model strengths on highly imbalanced pools; warm-starting reputations from pre-measured benchmark performance is a natural extension.