ARC: FAIR RELATIVE ADVANTAGE COMPARISON IN OPEN-ENDED REAL-WORLD INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-ended real-world interaction admits multiple valid behaviors: an agent may answer directly, ask for clarification, provide progress updates, or confirm before acting. This flexibility breaks a core assumption behind group-based RL: rollouts compared within a group are no longer guaranteed to be behaviorally comparable. As a result, reward-model preferences over interaction style can distort relative advantages and steer optimization toward reward-preferred behaviors rather than context-appropriate ones. We formalize this as a *reward fairness problem* and propose **ARC** (Advantage Regularization via Conditioning), a training recipe that restores fairer relative comparison through strategy-conditioned rollout grouping, together with hybrid rewards and entropy regularization. We study ARC in our proposed INTER^3, a novel paradigm for responsive, steerable, and execution-aware user-agent interaction that decouples user-visible communication from latent reasoning and tool use. INTER^3 also provides the annotation and distillation pipeline for constructing INTER^3-86K, our strategy-annotated training corpus for supervised and RL training. Empirically, ARC substantially strengthens the core τ/τ^2 tool-use benchmarks, while INTER^3 reduces time-to-first-token from 4.91s to 1.27s relative to a think-style baseline. Together, these results suggest that a central bottleneck in open-ended interactive learning is not only how agents are rewarded, but whether their behaviors are compared fairly in the first place. We release our ARC implementation and INTER^3-86K training data with this paper.

1 INTRODUCTION

Group-based RL and RLHF learn from relative rewards within sampled groups and are standard for language-model and agent post-training (Schulman et al., 2017; Ahmadian et al., 2024; Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b). Their signal is most interpretable when the compared rollouts occupy a locally comparable response region, so that centered rewards primarily reflect quality rather than qualitatively different behavioral modes. Yet reward models favor spurious attributes, including response length and style, which RL policies can exploit (Chen et al., 2024; Duan et al., 2026). When a rollout group spans sharply different behaviors, these biases need not be uniform across the group: samples may fall in different regions of the reward model's preference landscape, so the centered reward can absorb both quality differences and region-specific reward preference. The comparison remains relative, but it is no longer fair.

In open-ended interaction, an agent may answer directly, clarify missing information, provide progress updates, or confirm before an irreversible action (Yao et al., 2023b; 2025; Barres et al., 2025). Multiple choices may be valid for the same prompt, depending on missing information, reversibility, and dialogue state (Du et al., 2024; Tong et al., 2026b). We call the resulting bias the **reward fairness problem**: cross-strategy comparisons distort group-relative advantages and favor reward-preferred interaction styles.

We propose **ARC** (**A**dvantage **R**egularization via **C**onditioning). ARC assigns each training example a strategy instruction, compares rollouts within that strategy-conditioned group, and updates the policy with hybrid rewards and entropy regularization (Figure 1). At inference, the instruction is removed and strategy selection is autonomous. Unlike hint-based RL, which uses auxiliary information to improve exploration or expose solution structure (Liao et al., 2026; Zhang et al., 2026; Yan et al., 2025; Zhan et al., 2026), ARC uses instructions to make reward comparisons fairer.

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.

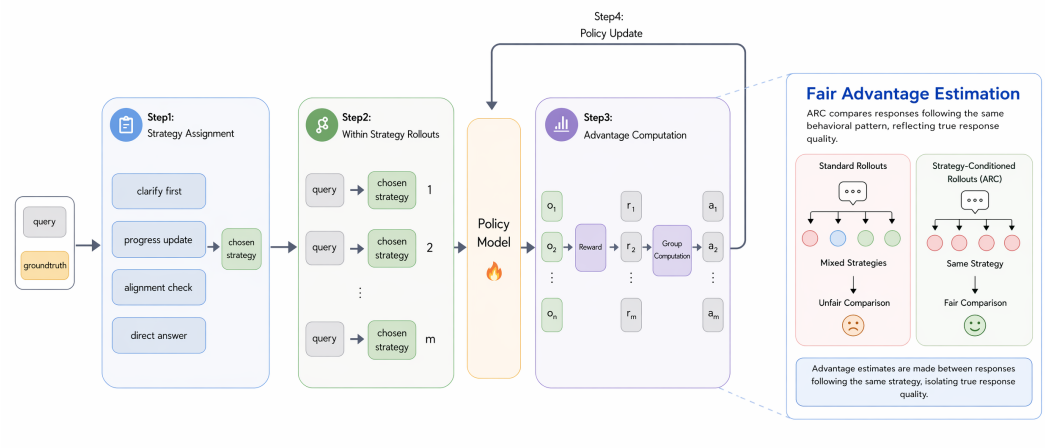

Figure 1: Overview of ARC. ARC modifies group-based RL by pairing each example with a strategy instruction during training, so rollouts are compared only within strategy-conditioned groups, yielding cleaner relative advantages by eliminating cross-strategy contamination. At inference, the instruction is removed and the policy selects strategies autonomously, making ARC a general recipe for fairer advantage estimation in open-ended interaction.

Existing agent frameworks capture task success more readily than the user experience: users may interrupt, redirect, pause, or lose patience while an agent works, and equally successful runs may feel very different (Yao et al., 2025; Barres et al., 2025; Vats et al., 2026; Liao & Vaughan, 2023; Liang et al., 2024). Think-then-act agents wait until reasoning and tool use finish before replying; ReAct agents place reasoning, actions, and replies in one visible stream (Yao et al., 2023b; Liang et al., 2025; Xie et al., 2026). Because both focus on model capability and verifiable outcomes, their trajectories reveal little about the different valid ways to communicate during execution.

We build **INTER^3** (Interplay of Internal Reasoning, Tool Usage, and Interaction), separating user-visible `<answer>` spans from latent reasoning and tool execution. Its system prompt defines ARC-aligned strategies for controllable, annotatable progress updates, clarification, alignment checks, and steering. The interface reduces time-to-first-token (TTFT) from 4.91s to 1.27s versus a think-then-act baseline. Online deployment traces, curated public data, and synthetic trajectories form **INTER^3-86K**, a strategy-annotated training corpus.

Ideal strategy conditioning removes an **inter-strategy variance** term from centered advantages. On INTER^3-86K, ARC improves agentic tool use; instruction diagnostics favor training with conditioning and inference without hints (Appendix K). INTER^3 exposes diverse behaviors, and ARC governs their comparison.

Our contributions are threefold:

- We propose **ARC**, a strategy-conditioned group-RL recipe for mitigating unfair advantage comparison in open-ended real-world agent interaction, and we analyze how it reduces **inter-strategy variance** in group-based advantage estimation.
- We build **INTER^3**, an async streaming interaction framework that separates user-visible communication from latent reasoning and tool execution, supports explicit strategy control and user interruption, and makes open-ended interaction observable for training.
- We construct **INTER^3-86K**, a strategy-annotated training substrate from real online traces, curated public data, and synthetic trajectories, and show it enables ARC to improve open-ended agentic RL while generalizing best to hint-free inference.

2 Related Work

Tool-augmented agents have made rapid progress in reasoning, planning, and external action (Schick et al., 2023; Qin et al., 2024; Yao et al., 2023b; Du et al., 2024; Qu et al., 2024; Tong et al., 2024b),

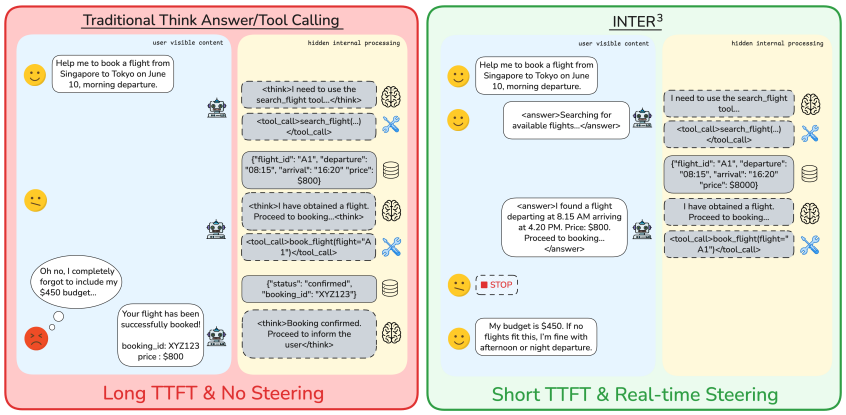

Figure 2: Standard think-then-answer (left) completes reasoning and tool execution before answering, resulting in high TTFT and no steering, whereas INTER^3 (right) interleaves reasoning, answering, and tool use, allowing real-time steering and faster TTFT.

while recent interleaving work has improved responsiveness by alternating between thought and partial output (Liang et al., 2025; Xie et al., 2026). Our departure is architectural: INTER^3 separates visible interaction from latent reasoning and tool execution, so users need not wait for the entire internal trajectory before the system can communicate. This turns responsiveness from a decoding behavior into an interface property, and makes strategy diversity operational in a realistic agent loop.

On the RL side, our work builds on policy-gradient and RLHF estimators (Schulman et al., 2017; Ahmadian et al., 2024; Rafailov et al., 2024; Ethayarajh et al., 2026), especially group-relative methods such as GRPO and its descendants (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b). These methods are effective when samples within a group are meaningfully comparable. Our focus is a different failure mode: in open-ended agent interaction, valid rollouts may differ in strategy rather than only in quality, so cross-strategy comparison can inject reward-model preferences directly into relative advantage estimates. ARC targets this comparability problem at rollout construction time.

The closest methodological neighbors are guidance-augmented RL approaches such as SAGE, Scaf-GRPO, LUFFY, and ExGRPO (Liao et al., 2026; Zhang et al., 2026; Yan et al., 2025; Zhan et al., 2026), which introduce auxiliary signals to improve exploration, mitigate sparse rewards, or reuse successful experience. ARC also uses an auxiliary instruction, but for a different reason: not to make the task easier, but to define a cleaner comparison class for group-relative optimization. Additional discussion of adjacent tool-use, interaction, and guidance literatures is deferred to Appendix B.

3 The INTER^3 Setting

INTER^3 separates interaction from execution: text outside `<answer>` is latent reasoning, tagged spans are streamed to users, and structured tool calls return results to context. Communication can therefore precede execution completion, and the same task can support different valid strategies. The model-agnostic interface requires adding `<answer>` tags to the tokenizer and extracting visible spans from generated text (Figure 2).

We implement INTER^3 in a lightweight asynchronous streaming runtime so users can receive updates and steer the agent before execution finishes. A single session supports internal reasoning, user-visible `<answer>` spans, tool calls, and interruptions. This enables responsive interaction while keeping communication distinct from internal execution (Figure 3).

Within this interface, the same task can admit several valid communication strategies. We organize them into four families: *Progress Update*, *Clarify First*, *Alignment Check*, and *Direct Answer*, which ARC uses as training-time comparison classes. The resulting **INTER^3-86K** corpus contains 86.8K examples (57.9K SFT, 28.9K RL) spanning tool use, multi-hop QA, and logical reasoning (Section 4). The runtime enables lower-latency communication; ARC improves learning within this setting.

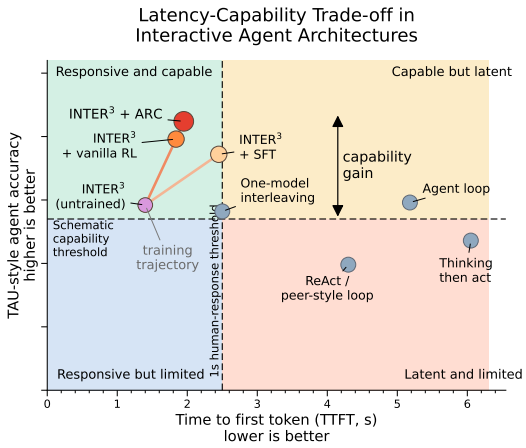

Figure 3: Latency-capability trade-off for interactive agent architectures. Channel-separated interaction shifts the operating point into a substantially lower-latency regime, while ARC improves capability *within* that regime rather than causing the horizontal TTFT shift itself. Non-INTER^3 baselines are schematic architectural placements rather than exact benchmark claims.

Table 1: INTER^3 strategy taxonomy used for ARC conditioning, analysis, and data construction.

Category	Strategy	Description
Progress Update	Tool Execute	Execute single or sequential tool calls; return the final answer upon completion
	Parallel Tools	Execute independent tool calls simultaneously; synthesize combined results
	Multi-Step Update	Stream incremental progress updates to the user between sequential sub-tasks
	Silent Execution	Execute tool calls without emitting user-visible output when intermediate steps are irrelevant
	Error Recovery	Handle tool failures gracefully and recover with corrective actions
Clarify First	Clarify First	Ask the user for missing or ambiguous information before executing any tool call
Alignment Check	Alignment Check	Restate the user's intent for confirmation before executing an irreversible action
	Decision Support	Present options and trade-offs for the user to make a judgment
Direct Answer	Direct Answer	Answer immediately from internal knowledge without invoking any tools

4 INTER^3-86K CONSTRUCTION

This section presents the construction of INTER^3-86K, including strategy taxonomy, data sources, curation, strategy annotation, and dataset statistics.

4.1 STRATEGY TAXONOMY

INTER^3 organizes interaction behavior into four high-level families and nine concrete strategies as seen in Table 1. ARC conditions rollout groups on the high-level family during training; the finer-grained strategies are useful for analysis, and interface documentation.

4.2 DATA SOURCES AND CURATION

We construct INTER^3-86K from two sources. The first comes from real-world deployment in customer service on a large-scale global payment platform under the INTER^3 runtime, where we collect online interaction traces exhibiting interruption, redirection, clarification, progress updates, and multi-step tool use. After de-identification and normalization into the INTER^3 format, these traces provide realistic open-ended interaction patterns that are difficult to recover from standard offline benchmarks alone.

The second source combines curated public benchmarks with teacher-driven augmentation, distillation, and synthesis. Public tool-use and reasoning data are rewritten into the INTER^3 format, expanded with strategy-conditioned variants, and supplemented with diversified synthetic interaction trajectories produced with a strong teacher model, Qwen3.5-397B-A17B (Qwen Team, 2026). The resulting SFT split contains 57.9K examples: 34.2K tool-use examples (59.1%) from the ToolMind collection,

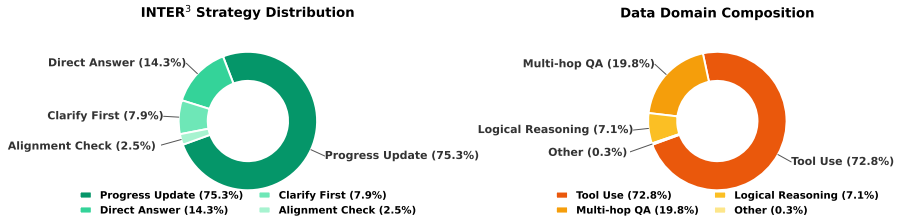

Figure 4: Left: INTER^3 strategy distribution in the RL training dataset. Progress Update dominates at 75.3%, reflecting the prevalence of multi-step tool-use scenarios that benefit from incremental user updates. Right: data domain composition of the full training set (SFT + RL). Tool Use accounts for 72.8% of examples, with Multi-hop QA (19.8%) and Logical Reasoning (7.1%) providing complementary reasoning diversity.

which aggregates public function-calling and agent benchmarks (Liu et al., 2024; 2025a; Yao et al., 2025; Zhang et al., 2025; Glaive AI, 2023; Ross et al., 2025; Chen et al., 2025b); 17.2K multi-hop QA examples (29.7%) from Musique Long Content (Trivedi et al., 2022); and 6.4K logical-reasoning or high-quality interleaved examples (11.1%) from KnightsAndKnaves (Xie et al., 2024) and Opus Distilled (TeichAI, 2025). The RL split contains 28.9K examples and is drawn entirely from tool-use data, to which we add strategy-conditioned prompts for ARC training. Figure 4 summarizes the resulting domain and strategy distributions.

4.3 SFT Data Construction

The SFT set draws from four data families with complementary reasoning demands. The largest component (∼34.2K, 59.1%) is the ToolMind collection, which aggregates seven publicly available tool-use benchmarks spanning diverse agentic scenarios. Since these benchmarks include ground-truth chain-of-thought annotations, we use a 397B LLM (Qwen3.5-397B-A17B (Qwen Team, 2026)) to rewrite each assistant turn into the INTER^3 format—interleaving internal reasoning with `<answer>` blocks and explicit tool calls—then filter malformed dialogues (invalid turn orderings, empty `<answer>` blocks). Three supplementary sources diversify the model's interleaved reasoning beyond tool calling: **KnightsAndKnaves** (∼6.2K, 10.7%) for multi-step deductive inference, **Musique Long Content** (∼17.2K, 29.7%) for multi-hop reasoning over extended passages, and **Opus Distilled** (∼250, 0.4%) for high-quality interleaved conversations generated via Claude Opus.

4.4 RL Data Construction

The RL set (∼28.9K) reuses the seven ToolMind sub-datasets but introduces training-time strategy instructions essential for ARC training. Each conversation undergoes a two-phase process: (1) a strategy-annotation pipeline assigns one *plausible* strategy instruction from our taxonomy—**Progress Update**, **Direct Answer**, **Clarify First**, or **Alignment Check**—based on the query, context, and response; (2) the selected instruction is injected into the system prompt, providing an explicit behavioral condition for *RL+ARC* training. These prompt-side strategy definitions are aligned with the high-level strategy families used by ARC, so the interaction interface, data annotation pipeline, and rollout comparison classes share the same control variables. In contrast, *baseline RL* training is performed on the same data without any injected strategy instruction.

4.5 Strategy Annotation

We annotate examples on the four high-level strategies—**Progress Update**, **Clarify First**, **Alignment Check**, and **Direct Answer**.

We employ a collaborative annotation approach using two large language models (Qwen3-235B-Instruct (Qwen Team, 2025) and Qwen3.5-27B (Qwen Team, 2026)) to ensure annotation quality and consistency. This dual-model verification captures the inherent subjectivity in strategy assignment while maintaining high inter-annotator agreement.

5 OUR METHODOLOGY: ADVANTAGE REGULARIZATION VIA CONDITIONING

While the INTER^3 setting makes interaction strategy observable, it does not solve the core RL challenge: how to compare responses fairly when they follow different communication patterns. We present **ARC** (**A**dvantage **R**egularization via **C**onditioning), a conditioning-based RL method for open-ended agent training. ARC's core mechanism is *strategy-conditioned rollout grouping*, which changes the comparison unit in group-based RL: relative advantages are computed within a strategy-conditioned subspace rather than across heterogeneous behaviors.

5.1 PROBLEM: UNFAIR ADVANTAGES IN MULTI-STRATEGY RL

In group-based RL, given prompt x, we sample N responses $\{y_1, \ldots, y_N\}$ from $\pi_\theta(\cdot|x)$ and compute advantages $\hat{A}_i = r_i - \bar{r}$, assuming comparable samples. When responses follow different strategies, this breaks: reward model bias contaminates the advantage signal, skewing policy updates toward reward-preferred behaviors regardless of task appropriateness.

Definition 1 (δ-Reward Fairness)**.** *A reward model $\mathcal{R}$ is δ-fair w.r.t. strategy set $\mathcal{S}$ if for any prompt x and responses y_i, y_j following different strategies with equal quality: $|\mathcal{R}(x, y_i) - \mathcal{R}(x, y_j)| \leq \delta$. A reward model is fair if $\delta = 0$, and unfair when $\delta > 0$ introduces systematic bias across strategies.*

In practice, reward models violate this fairness property due to exposure, length, and style bias—especially pronounced in agent settings where interaction appropriateness lacks clear ground truth.

5.2 STRATEGY-CONDITIONED ADVANTAGE ESTIMATION

The following results analyze a stylized prompt-conditional reward decomposition of the form $r = \mu_s(x) + \epsilon$. They isolate one source of estimator variance in group-relative advantages and should not be read as end-to-end convergence guarantees for GRPO. This estimator-centric viewpoint is closer to classical variance-reduction analyses for policy gradients (Greensmith et al., 2004; Yuan et al., 2022) and recent RLHF-side discussions of estimator behavior (Ahmadian et al., 2024; Liu et al., 2025b) than to a full convergence theory. For clarity, the theoretical comparison uses equal group size across the conditioned and unconditioned cases.

Theorem 5.1 (Idealized Variance Amplification)**.** *Under standard group-based RL sampling, the advantage variance is $\mathrm{Var}[\hat{A}_i] = (\sigma^2_{inter} + \sigma^2_{intra})(1 - 1/N)$, where σ^2_{intra} captures within-strategy noise and σ^2_{inter} captures between-strategy variance from reward model bias.*

The inter-strategy component σ^2_{inter} *persists* under standard sampling: no amount of oversampling can eliminate it. Our solution is to condition each rollout group on a specific strategy.

Definition 2 (Strategy-Conditioned Sampling)**.** *For each training example, we construct a strategy-conditioned prompt p^* by attaching a strategy instruction s^* from our interaction taxonomy, then sample M responses from $\pi_\theta(\cdot|x, s^*)$.*

Theorem 5.2 (Idealized Variance Reduction via Conditioning)**.** *Under strategy-conditioned sampling with the same group size: $\mathrm{Var}[\hat{A}_i|s^*] = \sigma^2_{intra} \cdot (1 - 1/N)$. Under perfect conditioning, the between-strategy mean-shift term is absent from this centered-advantage variance decomposition.*

Intuition. Heuristically, strategy conditioning emphasizes within-strategy reward dependence $I(r; y \mid x, s)$ while suppressing the strategy-linked term $I(r; s \mid x)$ that can contaminate cross-strategy comparison. Unlike hint-based RL methods (Liao et al., 2026; Zhang et al., 2026) that use hints to guide toward correct answers, we use strategy instructions to enforce cleaner within-strategy comparisons across diverse behaviors. See Appendix A for proofs and analysis.

Training-time strategy instructions do introduce a train-inference mismatch because they are removed at deployment. We study progressive instruction removal empirically in Section K.1 as a further analysis rather than as part of ARC itself.

5.3 Entropy Regularization for Multi-Channel Generation

The ARC training objective combines strategy-conditioned policy gradients with an entropy bonus:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{M} \hat{A}_i^{(s^*)} \cdot \log \pi_\theta(y_i^{(s^*)}|x, s^*) - \beta \cdot H(\pi_\theta(\cdot|x, s^*)) \,. \tag{1}$$

Why entropy regularization is critical for INTER^3. The interleaved output format requires balancing three channels—internal reasoning, user-facing <answer> tags, and tool calls. Without the entropy bonus ($\beta = 0$), we observe *entropy collapse* (Jin et al., 2026): the policy converges to emitting redundant <answer> blocks that receive marginal format rewards but carry no meaningful content. The entropy bonus counteracts this collapse by maintaining stochasticity across all output channels, encouraging diverse exploration of valid interleaving patterns.

5.4 Idealized Sample-Efficiency View

Theorem 5.3 (Idealized Gradient-Sample Scaling)**.** *Under the same stylized assumptions, if gradient-estimation error scales with the second moment of the score-function estimator, then the rollout requirement to estimate $\nabla J(\theta)$ up to accuracy ε scales as $n_{std} = O\Big(\frac{(\sigma^2_{intra}+\sigma^2_{inter})\log(1/\delta)}{\varepsilon^2}\Big)$ for standard GRPO and $n_{ARC} = O\Big(\frac{\sigma^2_{intra}\log(1/\delta)}{\varepsilon^2}\Big)$ for ARC.*

Corollary 1. Under the same assumptions, the implied sample-efficiency ratio is $n_{\text{std}}/n_{\text{ARC}} = 1 + \sigma^2_{\text{inter}}/\sigma^2_{\text{intra}}$. This is best read as an idealized scaling comparison rather than a full convergence guarantee, in the same spirit as estimator-level policy-gradient analyses (Greensmith et al., 2004). We use it as an estimator-level interpretation of why cleaner within-strategy comparison can reduce the gradient-sampling burden, rather than as a calibrated empirical speedup claim. See Appendix A for the derivation and assumptions.

5.5 Implementation

ARC consists of four steps (Figure 1):

1. **Strategy Instruction Assignment**: each training example is paired with one strategy instruction s^* from the interaction taxonomy.
2. **Within-Strategy Sampling**: sample M rollouts from $\pi_\theta(\cdot \mid x, s^*)$.
3. **Advantage Computation**: compute advantages within each strategy group.
4. **Policy Update**: update the policy using the entropy-regularized objective.

At inference, no strategy instruction is provided; the model autonomously selects appropriate strategies. Section K.1 further analyzes instruction-removal curricula as an auxiliary experiments.

6 Experiments

6.1 Experimental Setup

Model and training. We use Qwen3-8B (Qwen Team, 2025) trained with GRPO (Shao et al., 2024) in no-think mode, warm-started from the same INTER^3 SFT checkpoint for all RL backbones. ARC's core mechanism is strategy-conditioned rollout grouping; in the full INTER^3 instantiation additionally apply entropy regularization as a stabilizer for multi-channel generation. Unless explicitly varied, all methods use the same final reward construction, selected based on ablations in Appendix C. Additional ablations on entropy regularization are provided in Appendix D.

Benchmarks. We evaluate on two dimensions: (1) *In-domain agentic capabilities* using tau-bench (Yao et al., 2025) and tau2-bench (Barres et al., 2025) for multi-turn tool calling in airline, retail, and telecom scenarios; (2) *Out-of-domain reasoning* using Arena-Hard (Li et al., 2025), AIME 2026 (Balunovic et al., 2025), GPQA-Diamond (Rein et al., 2024), IFBench (Pyatkin et al., 2025), and HMMT (Balunovic et al., 2025).

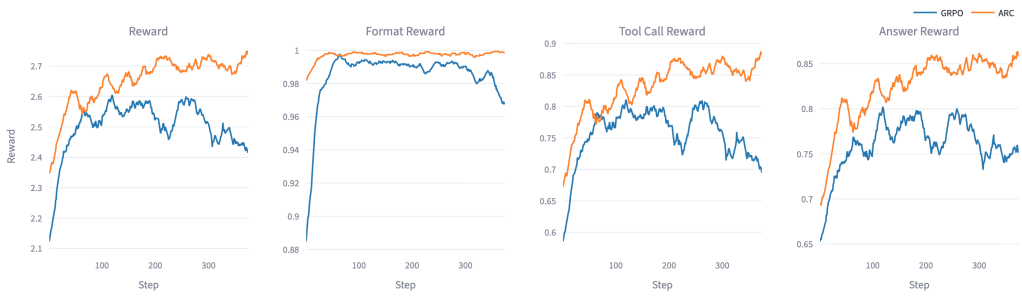

Figure 5: Training reward curves for GRPO and ARC.

Table 2: Main results. ARC delivers clearest gains on in-domain tool use, especially on GRPO. Avg. is the mean over displayed metrics; red parentheses show Avg. change vs. backbone. TTFT (↓) averages available τ-bench latency measurements.

Method	Avg.	TTFT ↓	Tool Calling					Reasoning			Instruction Following	Alignment
			τ-airline	τ-retail	τ^2-airline	τ^2-retail	τ^2-telecom	AIME 2026	GPQA-D	HMMT 2025	IFBench	ArenaHard
Baselines												
Qwen3-8B-noThink	22.85	0.05s	12.00	29.86	14.61	36.55	17.80	17.08	45.71	12.92	24.83	17.14
Qwen3-8B-Think	32.82	4.91s	28.00	36.81	29.75	38.71	23.46	47.92	53.54	20.83	19.43	29.83
RL Backbones												
PPO	27.49	0.45s	35.33	41.45	33.33	38.89	19.01	30.83	38.70	8.75	19.73	8.88
PPO + ARC (Ours)	28.57 (+1.08)	0.78s	39.33	46.09	41.61	44.44	20.76	19.17	41.10	10.00	14.29	8.92
DAPO	28.61	0.62s	31.33	42.32	35.33	42.11	19.37	36.67	39.02	13.33	12.24	14.36
DAPO + ARC (Ours)	29.92 (+1.31)	0.82s	34.67	41.74	34.50	40.64	19.64	41.25	43.94	12.50	17.35	12.92
GRPO	28.09	0.61s	31.33	40.29	36.67	40.64	17.84	31.67	40.91	9.17	18.71	13.66
GRPO + ARC (Ours)	33.46 (+5.37)	1.27s	44.00	50.00	48.00	45.61	21.05	40.83	41.41	12.50	15.99	15.18

Baseline methods. We benchmark ARC against two classes of baselines: (1) *Qwen3-8B-noThink / Think*, minimal reasoning variants to quantify ARC's gains; (2) *Standard RL backbones* (PPO (Schulman et al., 2017), DAPO (Yu et al., 2025), GRPO (Shao et al., 2024)), representing widely-used reinforcement learning methods.

6.2 Main Results

In-domain performance. Table 2 shows that ARC consistently improves performance across all RL backbones, with the largest gains for GRPO (+5.37 average points). Improvements are especially pronounced in in-domain tool-calling tasks: for GRPO, τ-airline rises from 31.33 to 44.00, τ-retail from 40.29 to 50.00, and τ^2-airline from 36.67 to 48.00, with smaller gains in τ^2-retail and τ^2-telecom. These results demonstrate ARC's ability to enhance multi-turn tool use and in-domain capabilities, particularly with stronger RL backbones.

Out-of-domain reasoning and tradeoffs. On out-of-domain reasoning, instruction following, and alignment benchmarks, ARC shows more nuanced effects. GRPO + ARC improves reasoning on AIME 2026 from 31.67 to 40.83 and provides modest gains in instruction-following and alignment metrics, while improvements for PPO and DAPO backbones are mixed, suggesting that ARC's benefits depend on the underlying optimization dynamics. These results show that while ARC's primary impact is on structured tool use, it can also unlock meaningful out-of-domain reasoning gains under suitable training regimes.

Training dynamics and robustness. Figure 5 shows that ARC's benefits extend beyond final performance to training dynamics. While GRPO peaks mid-training and then declines—especially in tool-call and answer rewards—ARC maintains or improves across the same budget. Since format reward saturates for both methods, this divergence reflects execution and answer quality rather than syntactic compliance. This suggests that standard group-based RL initially learns useful behaviors but then exploits reward model biases, whereas ARC's strategy-conditioned comparison prevents degradation by preserving fair advantage signals throughout training.

Table 3: Qwen3-4B case study. ARC remains effective at smaller scale and outperforms no-think, think, and 4B GRPO baselines on the in-domain tool-use suite.

Method	Avg.	τ airline	τ retail	τ^2 airline	τ^2 retail	τ^2 telecom
Qwen3-4B-noThink	22.38	20.00	22.32	21.40	26.31	21.90
Qwen3-4B-Think	29.54	27.51	37.10	28.07	29.21	25.83
GRPO (4B)	22.33	29.33	10.72	36.94	15.96	18.69
GRPO+ARC (4B)	34.23	29.33	40.00	32.00	39.77	30.03

6.3 Small-Model Case Study

Table 3 shows that the agentic benefit of ARC is not specific to the 8B scale. Even at 4B, ARC remains stronger than no-think, think, and the corresponding 4B GRPO baseline on the tool-use suite, increasing the five-task average by about 53% over 4B no-think. This finding suggests that ARC improves performance by strengthening the training comparison signal in agentic settings, rather than simply leveraging larger model capacity.

7 Conclusion

Open-ended agent interaction often admits multiple valid behaviors, yet standard group-based RL compares rollouts within a shared relative-reward pool. We argue this creates a reward fairness issue: advantage estimates become confounded when heterogeneous strategies are normalized together rather than reflecting intrinsic quality differences.

ARC addresses this by conditioning the comparison class during rollout construction so that each group contains a single strategy family. This improves the interpretability of relative advantages in multi-strategy regimes without altering the underlying policy optimization machinery. Complementarily, the INTER^3 setting exposes this issue in practice by separating user-facing communication from latent reasoning and tool execution, revealing diverse valid strategies within the same task family.

Empirically, ARC is most effective on in-domain agentic benchmarks where multi-strategy interaction is prevalent. Training on INTER^3-86K yields higher and more stable rewards, including reduced post-peak degradation under fixed compute budgets. However, gains are not uniform across backbones or downstream metrics, suggesting ARC is a targeted correction for comparison bias rather than a universally dominant optimizer.

Our theoretical analysis provides a mechanism-level explanation for these effects. The results in Section 5 are stylized estimator-level characterizations, not end-to-end convergence guarantees. They show how inter-strategy variance in reward-model evaluations can reduce sample efficiency, and how restricting the comparison class can mitigate this variance under strategy-dependent reward bias.

Overall, progress in open-ended agent learning may depend not only on stronger models or reward signals, but also on constructing fair comparison sets when multiple valid behaviors coexist.

Limitations

Our study has four main limitations.

(1) Strategy abstraction. ARC relies on a coarse strategy taxonomy. While useful for conditioning and analysis, real interaction behaviors are more nuanced and context-dependent (Lin et al., 2023; Pan et al., 2025).

(2) Domain scope. Our strongest results are in open-ended tool-usage settings. Whether ARC generalizes to other domains remains to be tested.

(3) Data and annotation dependence. The method depends on normalized interaction traces and strategy labels, which may reflect annotation bias or deployment-specific patterns.

(4) Theoretical scope. Variance and sample-efficiency analyses clarify one optimization mechanism but do not constitute a full convergence theory for open-ended agent RL.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. In *Advances in Neural Information Processing Systems*, 2025.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024.

Mengyuan Chen, Chengjun Dai, Xinyang Dong, Chengzhe Feng, Kewei Fu, Jianshe Li, Zhihan Peng, Yongqi Tong, Junshao Zhang, and Hong Zhu. Dingtalk deepresearch: A unified multi agent framework for adaptive intelligence in enterprise environments. *arXiv preprint arXiv:2510.24760*, 2025a. URL https://arxiv.org/abs/2510.24760.

Mingyang Chen, Haoze Sun, Tianpeng Li, Fan Yang, Hao Liang, Keer Lu, Bin Cui, Wentao Zhang, Zenan Zhou, and Weipeng Chen. Facilitating multi-turn function calling for llms via compositional instruction tuning. In *International Conference on Learning Representations*, 2025b.

José Antonio Siqueira de Cerqueira, Mamia Agbese, Rebekah Rousi, Nannan Xi, Juho Hamari, and Pekka Abrahamsson. Can we trust ai agents? a case study of an llm-based multi-agent system for ethical ai. *arXiv preprint arXiv:2411.08881*, 2026.

DeepSeek-AI. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 2025.

Yu Du, Fangyun Wei, and Hongyang Zhang. Anytool: Self-reflective, hierarchical agents for large-scale API calls. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=qFILbkTQWw.

Zhibin Duan, Guowei Rong, Zhuo Li, Bo Chen, Mingyuan Zhou, and Dandan Guo. Mitigating reward hacking in rlhf via bayesian non-negative reward modeling. *arXiv preprint arXiv:2602.10623*, 2026.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2026.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents, 2018a. URL https://arxiv.org/abs/1705.06366.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning, 2018b. URL https://arxiv.org/abs/1707.05300.

Glaive AI. Glaive function calling v2. https://huggingface.co/datasets/glaiveai/glaive-function-calling-v2, 2023. Hugging Face dataset.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Sil Hamilton. Detecting mode collapse in language models via narration. *arXiv preprint arXiv:2402.04477*, 2024.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. *arXiv preprint arXiv:2511.05993*, 2026.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Anthony Liang, Jonathan Berant, Adam Fisch, Abhimanyu Goyal, Kalpesh Krishna, and Jacob Eisenstein. Plantain: Plan-answer interleaved reasoning. *arXiv preprint arXiv:2512.03176*, 2025.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*, 2024.

Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-hinting language models enhance reinforcement learning. *arXiv preprint arXiv:2602.03143*, 2026.

Q. Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 2023.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user-AI conversation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4694–4702, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.311. URL https://aclanthology.org/2023.findings-emnlp.311/.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. Toolace: Winning the points of llm function calling. In *International Conference on Learning Representations*, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. In *Advances in Neural Information Processing Systems*, 2024.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *arXiv preprint arXiv:1707.00183*, 2017.

Licheng Pan, Yongqi Tong, Xin Zhang, Xiaolu Zhang, Jun Zhou, and Zhixuan Chu. Understanding and mitigating overrefusal in LLMs from an unveiling perspective of safety decision boundary. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21057–21075. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1065. URL https://aclanthology.org/2025.emnlp-main.1065/.

Licheng Pan, Haochen Yang, Haoxuan Li, Yunsheng Lu, Yongqi Tong, Yinuo Wang, Shijian Wang, Zhixuan Chu, Lei Shen, Yuan Lu, and Hao Wang. Optimal transport for llm reward modeling from noisy preference. *arXiv preprint arXiv:2605.06036*, 2026. URL https://arxiv.org/abs/2605.06036.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis. In *Advances in Neural Information Processing Systems*, 2024.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, 2024.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey. *arXiv preprint arXiv:2405.17935*, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 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, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 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.

Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. When2Call: When (not) to call tools. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

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

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Sharon Li, Jason E Weston, and Ping Yu. Hybrid reinforcement: When reward is sparse, it's better to be dense. *arXiv preprint arXiv:2510.07242*, 2025.

TeichAI. Claude 4.5 opus high reasoning 250x dataset. https://huggingface.co/datasets/TeichAI/claude-4.5-opus-high-reasoning-250x, 2025.

Yongqi Tong, Yifan Wang, Dawei Li, Sizhe Wang, Zi Lin, Simeng Han, and Jingbo Shang. Eliminating reasoning via inferring with planning: A new framework to guide LLMs' non-linear thinking. *arXiv preprint arXiv:2310.12342*, 2023. URL https://arxiv.org/abs/2310.12342.

Yongqi Tong, Dawei Li, Sizhe Wang, Yujia Wang, Fei Teng, and Jingbo Shang. Can LLMs learn from previous mistakes? investigating LLMs' errors to boost for reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3065–3080. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.169. URL https://aclanthology.org/2024.acl-long.169/.

Yongqi Tong, Sizhe Wang, Dawei Li, Yifan Wang, Simeng Han, Zi Lin, Chengsong Huang, Jiaxin Huang, and Jingbo Shang. Optimizing language model's reasoning abilities with weak supervision. *arXiv preprint arXiv:2405.04086*, 2024b. URL https://arxiv.org/abs/2405.04086.

Yongqi Tong, Xiaoyun Feng, Lyuxin Xue, Jianshe Li, Xin Zhang, and Jiang-Ming Yang. How to train a real-world silicon concierge? internalizing complex business workflow to only OneModel. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1303–1316. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-industry.91. URL https://aclanthology.org/2026.acl-industry.91/.

Yongqi Tong, Zhenyu Zhang, Zimi Liu, Tangzeyun, Zhihan Peng, Kewei Fu, Mingli Song, Haofei Zhang, Junshao Zhang, Hong Zhu, and JIANSHE LI. Reinforcement learning with missing context to mitigate reward hacking from training only on golden answers, 2026b. URL https://openreview.net/forum?id=eOqnXPOl4S. OpenReview preprint.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.

Vanshika Vats, Marzia Binta Nizam, Minghao Liu, Ziyuan Wang, Richard Ho, Mohnish Sai Prasad, Vincent Titterton, Sai Venkat Malreddy, Riya Aggarwal, Yanwen Xu, Lei Ding, Jay Mehta, Nathan Grinnell, Li Liu, Sijia Zhong, Devanathan Nallur Gandamani, Xinyi Tang, Rohan Ghosalkar, Celeste Shen, Rachel Shen, Nafisa Hussain, Kesav Ravichandran, and James Davis. A survey on human-ai collaboration with large foundation models. *ACM Transactions on Intelligent Systems and Technology*, August 2026. ISSN 2157-6912. doi: 10.1145/3841472. URL http://dx.doi.org/10.1145/3841472.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhi-Qiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:258558102.

Sizhe Wang, Yongqi Tong, Hengyuan Zhang, Dawei Li, Xin Zhang, and Tianlong Chen. BPO: Towards balanced preference optimization between knowledge breadth and depth in alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8811–8826. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.443. URL https://aclanthology.org/2025.naacl-long.443/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. *arXiv preprint arXiv:2410.23123*, 2024.

Roy Xie, David Qiu, Deepak Gopinath, Dong Lin, Yanchao Sun, Chong Wang, Saloni Potdar, and Bhuwan Dhingra. Interleaved reasoning for large language models via reinforcement learning. *arXiv preprint arXiv:2505.19640*, 2026.

Zhihao Xu, Yongqi Tong, Xin Zhang, Jun Zhou, editor = "Liakata Maria Wang, Xiting", Viviane P. Moreira, Jiajun Zhang, and David Jurgens. Understanding conflicts in multi-objective alignment through reward consistency. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5450–5472. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.269. URL https://aclanthology.org/2026.findings-acl.269/.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *Advances in Neural Information Processing Systems*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025.

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, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R. Fung, Hao Peng, and Heng Ji. Craft: Customizing llms by creating and retrieving from specialized toolsets, 2024. URL https://arxiv.org/abs/2309.17428.

Rui Yuan, Robert M. Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. *arXiv preprint arXiv:2107.11433*, 2022.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. Exgrpo: Learning to reason from experience. In *International Conference on Learning Representations*, 2026.

Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Zhiwei Liu, Yihao Feng, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Silvio Savarese, and Caiming Xiong. xLAM: A family of large action models to empower AI agent systems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. In *International Conference on Learning Representations*, 2026.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2024. URL https://arxiv.org/abs/2310.04406.

A Proofs and Theoretical Analysis

This appendix provides proofs for the stylized variance claims in Section 5. We distinguish between the *target strategy* $s^\star$, which is assigned to a training prompt before ARC rollout generation, and the *realized strategy* $s_i = f(y_i)$ exhibited by a sampled response y_i.

$$\Pr_{y\sim\pi_\theta(\cdot|x)}[f(y) = s]. \tag{2}$$

Under ARC, a target strategy $s^\star$ is assigned before rollout generation and appended to the prompt. Responses are then sampled as

$$y_i \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x, s^\star). \tag{3}$$

The realized strategy remains $s_i = f(y_i)$ and may differ from $s^\star$ when strategy adherence is imperfect.

A.1 Proof of Theorem 5.1 (Variance Amplification)

Proof. Decompose $r_i = \mu_{s_i} + \epsilon_i$ where μ_{s_i} is the mean reward for realized strategy s_i (capturing RM bias) and ϵ_i is i.i.d. zero-mean noise with variance σ^2_{intra}, independent of the strategy-dependent mean term. Since $y_i \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x)$ and $s_i = f(y_i)$, the realized strategies $s_1, \ldots, s_N$ are i.i.d. under the induced distribution $P_\theta(\cdot \mid x)$, so $\mu_{s_1}, \ldots, \mu_{s_N}$ are i.i.d. with variance σ^2_{inter}.

The advantage is $\hat{A}_i = r_i - \bar{r} = (\mu_{s_i} - \bar{\mu}) + (\epsilon_i - \bar{\epsilon})$, where $\bar{\mu} = \frac{1}{N}\sum_j \mu_{s_j}$ and $\bar{\epsilon} = \frac{1}{N}\sum_j \epsilon_j$.

Since strategy and noise are independent:

$$\mathrm{Var}[\hat{A}_i] = \mathrm{Var}[\mu_{s_i} - \bar{\mu}] + \mathrm{Var}[\epsilon_i - \bar{\epsilon}] \tag{4}$$

For any i.i.d. sequence $Z_1, \ldots, Z_N$ with variance σ^2:

$$\mathrm{Var}[Z_i - \bar{Z}] = \left(1 - \frac{1}{N}\right)^2 \sigma^2 + \frac{N-1}{N^2}\sigma^2 = \sigma^2\left(1 - \frac{1}{N}\right) \tag{5}$$

Applying this to both terms yields $\mathrm{Var}[\hat{A}_i] = (\sigma^2_{\text{inter}} + \sigma^2_{\text{intra}})(1 - 1/N)$. □

A.2 Proof of Theorem 5.2 (Variance Reduction via Conditioning)

Proof. ARC conditions generation on the target strategy $s^\star$, and under perfect compliance, $f(y_i) = s^\star$, so $\mu_{s_i} = \mu_{s^*}$ for all i. The advantage simplifies to:

$$\hat{A}_i = \epsilon_i - \bar{\epsilon} \tag{6}$$

Since ϵ_i are i.i.d. with variance σ^2_{intra}:

$$\mathrm{Var}[\hat{A}_i | s^*] = \sigma^2_{\text{intra}}\left(1 - \frac{1}{N}\right) \tag{7}$$

The between-strategy mean-shift term is absent from this centered-advantage variance decomposition.

$$\begin{aligned}\mathrm{Var}[\hat{A}_i | s^*] = \Big(&\sigma^2_{\text{intra}}(x, s^\star) \\ &+ \mathrm{Var}_{S\sim P\theta(\cdot|x,s^\star)}[\mu_S(x)]\Big)\left(1 - \frac{1}{N}\right).\end{aligned} \tag{8}$$

With imperfect compliance, the ARC variance additionally contains the residual between-strategy term, so the predicted variance reduction is correspondingly attenuated. □

A.3 Derivation for Theorem 5.3 (Idealized Sample-Complexity Comparison)

Theorem A.1 (Gradient Variance Bound)**.** *Let g_{ARC} and g_{std} denote policy gradients under strategy-conditioned and standard sampling with equal group size N. For analytic tractability, assume: (i) the relevant advantage-score mixed moments approximately factorize conditional on x; (ii) the score-function second moments $F = \mathbb{E}[\|\nabla_i\|^2]$ and $F' = \mathbb{E}[\langle \nabla_i, \nabla_j \rangle]$ $(i \neq j)$ are comparable up to constants across the two sampling schemes; and (iii) ARC satisfies the perfect-compliance idealization of Theorem 5.2. Then:*

$$\frac{\mathbb{E}[\|g_{ARC}\|^2]}{\mathbb{E}[\|g_{std}\|^2]} \approx \frac{\sigma^2_{intra}}{\sigma^2_{intra} + \sigma^2_{inter}} \tag{9}$$

Proof. Write $g = \frac{1}{N}\sum_{i=1}^{N} \hat{A}_i \nabla_i$ where $\nabla_i = \nabla_\theta \log \pi_\theta(y_i|x)$. Under assumption (i):

$$\mathbb{E}[\|g\|^2] = \frac{1}{N^2}\left[N\mathbb{E}[\hat{A}_i^2]\, F + N(N-1)\,\mathbb{E}[\hat{A}_i \hat{A}_j]\, F'\right] \tag{10}$$

For i.i.d. rewards r_i with total variance σ_r^2, the centered advantages satisfy:

$$\mathbb{E}[\hat{A}_i^2] = \sigma_r^2\left(1 - \frac{1}{N}\right) \tag{11}$$

$$\mathbb{E}[\hat{A}_i \hat{A}_j] = -\frac{\sigma_r^2}{N} \quad (i \neq j) \tag{12}$$

The latter follows from $\mathrm{Cov}[r_i - \bar{r},\, r_j - \bar{r}] = -\sigma_r^2/N$.

Substituting:

$$\mathbb{E}[\|g\|^2] = \frac{\sigma_r^2(N-1)}{N^2}(F - F') \tag{13}$$

Since $\sigma_r^2 = \sigma^2_{\text{intra}} + \sigma^2_{\text{inter}}$ under standard sampling and $\sigma_r^2 = \sigma^2_{\text{intra}}$ under ARC (Theorems 5.1–5.2), assumption (ii) yields the stated approximation. □

Derivation for Theorem 5.3. This argument should be read as a stylized variance-to-sample-efficiency translation rather than a full convergence proof. Under a sub-Gaussian gradient-estimation assumption, the number of gradient samples required to achieve $\|\hat{g} - \nabla J(\theta)\| \leq \varepsilon$ with probability $\geq 1 - \delta$ scales as $O(\sigma_g^2 \log(1/\delta)/\varepsilon^2)$, where σ_g^2 is the per-sample gradient variance. From Theorem A.1, $\sigma^2_{g,\text{ARC}}/\sigma^2_{g,\text{std}} \approx \sigma^2_{\text{intra}}/(\sigma^2_{\text{intra}} + \sigma^2_{\text{inter}})$, and the same approximation carries over to this idealized sample-efficiency comparison. □

A.4 Effective Variance Analysis for Curriculum Learning

For the curriculum learning schedule studied in Section K.1, the expected advantage variance across the batch at training step t is a mixture of conditioned and unconditioned groups:

$$\begin{aligned}\mathrm{Var}_{\text{eff}}(t) = \big(1 - p_d(t)\big)\,\sigma^2_{\text{intra}}\big(1 - \tfrac{1}{N}\big) \\ + p_d(t)\,\big(\sigma^2_{\text{intra}} + \sigma^2_{\text{inter}}\big)\big(1 - \tfrac{1}{N}\big)\,.\end{aligned} \tag{14}$$

Since the second term exceeds the first by $\sigma^2_{\text{inter}}(1 - 1/N)$, $\mathrm{Var}_{\text{eff}}(t)$ increases monotonically from $\approx \sigma^2_{\text{intra}}(1-1/N)$ to $\approx (\sigma^2_{\text{intra}} + \sigma^2_{\text{inter}})(1-1/N)$. This monotonic increase gives the curriculum learning schedule a variance-based progression: the model trains under progressively noisier optimization landscapes as it becomes more capable.

A.5 Information-Theoretic Perspective

Assuming strategy s is determined by the response y (i.e., $s = f(y)$ for some deterministic function), the chain rule of mutual information gives:

$$I(r; y|x) = I(r; y|x, s) + I(r; s|x) \tag{15}$$

The term $I(r; s|x)$ captures strategy-linked reward dependence that is orthogonal to within-strategy quality assessment.

Under the idealized perfect-compliance setting, conditioning on the target strategy removes between-strategy variation from within-group comparisons, emphasizing the within-strategy reward signal $I(r; y \mid x, s)$.

B Extended Related Work

Tool-Augmented and Interactive Agents. Tool-augmented language models have been studied through self-supervised tool learning (Schick et al., 2023), large-scale API integration (Qin et al., 2024; Patil et al., 2024), synthetic function-calling corpora (Liu et al., 2024; 2025a), and increasingly sophisticated agent architectures (Du et al., 2024; Yuan et al., 2024; Chen et al., 2025a; Tong et al., 2026a). A parallel line of work studies interaction structure, including ReAct-style reasoning-action interleaving (Yao et al., 2023b), structured planning (Wei et al., 2022; Yao et al., 2023a; Wang et al., 2023; Tong et al., 2023), and reflective or search-based agents (Shinn et al., 2023; Zhou et al., 2024; Tong et al., 2024a). Our interest is not only in adding tools or planning steps, but in exposing interaction as a first-class, user-visible channel while execution is still ongoing.

Interleaved Reasoning and User Experience. Recent interleaving methods improve perceived responsiveness by alternating internal reasoning with partial textual output (Liang et al., 2025; Xie et al., 2026). These works mainly address the sequencing of thought and answer tokens. INTER^3 targets a different bottleneck: long-running external actions. When a tool call itself is the latency source, making the communication channel independent from the execution channel becomes the key design move. This connects to broader human-AI interaction work on transparency, feedback, and controllability (Vats et al., 2026; Liao & Vaughan, 2023; de Cerqueira et al., 2026; Liang et al., 2024; Li et al., 2025), but in an agent context where the cost of opaque waiting is especially high.

RLHF Estimators and Multi-Behavior Collapse. Our RL analysis sits within the broader literature on policy-gradient and RLHF optimization (Schulman et al., 2017; Ahmadian et al., 2024; Rafailov et al., 2024; Ethayarajh et al., 2026; Wang et al., 2025; Xu et al., 2026; Pan et al., 2026), especially group-relative estimators such as GRPO (Shao et al., 2024) and refinements for entropy preservation or bias correction (Yu et al., 2025; Liu et al., 2025b). These methods have proven effective for reasoning-heavy tasks (DeepSeek-AI, 2025), but open-ended multi-behavior settings raise a distinct collapse risk (Hamilton, 2024): the policy can converge to whichever behavior the reward favors. Our contribution is to isolate one concrete statistical mechanism for that collapse—cross-strategy contamination of relative advantages—and study a conditioning-based remedy.

External Guidance in RL. Methods such as SAGE (Liao et al., 2026), Scaf-GRPO (Zhang et al., 2026), LUFFY (Yan et al., 2025), and ExGRPO (Zhan et al., 2026) show that auxiliary guidance can materially improve RL training by mitigating sparse rewards, structuring exploration, incorporating demonstrations, or reusing successful trajectories. ARC is adjacent in form but different in purpose. The strategy instruction is not introduced to reveal the answer, densify reward, or bias the policy toward a target trajectory. It is introduced to constrain which rollouts are compared to one another, so that relative advantage estimation is performed inside a behaviorally coherent comparison class.

C Reward Details

C.1 Final Reward Used in Main Experiments

The specific reward used in our full INTER^3 instantiation is not ARC's core mechanism; it is a setting-specific design choice for stabilizing execution-grounded training. Following recent reward-design work that combines verifiable structural signals with denser model-based feedback (Qian et al., 2025; Tao et al., 2025), we use a compact three-term reward. The **format reward** $R_{\text{fmt}} \in \{0, 1\}$ checks for at least one well-formed `<tool_call>` or `<answer>` block with properly matched, non-nested tags. The **tool reward** $R_{\text{tool}} \in \{-1, 0, 1\}$ evaluates exact tool-call correctness: the tool identifier, argument key set, and argument values must all match the reference for $R_{\text{tool}} = 1$. The **answer reward** $R_{\text{ans}} \in \{0, 0.50, 1\}$ is an LLM-judge score on the extracted `<answer>` content. To couple semantics to correct execution, we set $R_{\text{ans}} = \text{Eval}(y, y^*)$ only when $R_{\text{tool}} = 1$, and $R_{\text{ans}} = 0$

Table 4: Reward-construction ablation. All variants include the format reward. Exact tool matching improves tool use, and the final gated design gives the strongest tool-centric operating point. Unless explicitly varied, all main RL baselines and ARC variants in the paper use the final row.

Reward Comparisons	Avg.	τ Avg.	τ^2 Avg.	Reasoning Avg.	IFBench	ArenaHard
Partial tool + answer judge	24.90	30.92	29.16	33.19	14.97	16.24
Exact tool + answer judge	27.89	37.31	31.94	36.24	18.03	15.94
Penalized tool + gated judge	27.98	43.93	36.49	29.60	16.33	13.57

Table 5: Reward-model ablation for the same Qwen3-8B policy under the final reward construction used throughout the main experiments. Varying only the external judge shows that the 235B judge provides the strongest tool-use operating point.

Judge Model	Avg.	τ Avg.	τ^2 Avg.	Reasoning Avg.	IFBench	ArenaHard
Qwen3-235B-Instruct	27.98	43.93	36.49	29.60	16.33	13.57
Qwen3.5-27B	21.67	25.69	22.55	27.97	19.39	12.77
Qwen3.5-122B	25.02	35.22	30.70	29.27	16.67	13.27

otherwise. The final reward is simply $R = R_{\text{fmt}} + R_{\text{tool}} + R_{\text{ans}}$. If both prediction and reference contain no answer spans, we set $R_{\text{ans}} = 1$ to reflect structural agreement; if evaluation fails, we return a neutral fallback score.

C.2 Detailed Reward Ablation

A central challenge in training INTER^3 agents is designing rewards that evaluate multi-channel outputs—internal reasoning, tool calls, and user-facing `<answer>` spans—without baking in strategy-dependent bias. We study two reward-side choices: the reward construction itself, and the reward model used to score semantic answer quality.

The reward constructions in Table 4 differ along two axes: (i) tool-execution strictness, and (ii) coupling between execution and semantic reward. All variants include a shared **format reward** enforcing structural validity. Moving from partial to exact tool matching strengthens the execution-grounded signal, and the final gated design amplifies this effect, increasing τ Avg. by about 42% over the weakest variant. This comes with reduced performance on broader reasoning-style evaluations, while leaving overall average performance largely unchanged. The results reflect our objective: to prioritize rewards that favor answers grounded in correct tool execution rather than signals agnostic to whether the answer is causally supported by the executed actions.

Table 5 fixes the policy at Qwen3-8B and varies only the external reward judge. Under this controlled comparison, the 235B judge gives the strongest τ/τ^2 performance and the best overall balance for execution-grounded training, while smaller judges recover only isolated gains on auxiliary metrics. These findings led us to select the 235B judge for our main experiments, as it provides the most consistent signal for tool-execution verification.

Tool matching. *Partial tool* assigns credit to approximately correct tool calls, providing dense but noisy supervision. Replacing this with **exact tool matching** yields stricter credit assignment, improving reliability of the training signal.

Negative penalties. The final variant introduces a **negative penalty** for incorrect tool execution, explicitly discouraging spurious or malformed calls and sharpening optimization toward valid trajectories.

Semantic reward coupling. All variants use an LLM-based **answer judge** for user-visible responses. In the first two variants, this reward is applied unconditionally, allowing fluent but unsupported answers to receive credit. The final variant applies a **gated answer judge**, where semantic reward is issued only when exact tool execution is correct, coupling answer quality with execution validity.

Table 6: Ablation study on ARC: Training settings and entropy hyperparameters. Strategy-conditioned grouping provides the largest single gain over base GRPO. While entropy regularization alone slightly decreases performance, combining it with strategy-conditioned grouping further improves the overall operating point. The best entropy value (0.001) yields the highest average score.

Configuration	Avg.	τ Avg.	τ^2 Avg.	Reasoning Avg.	IFBench	ArenaHard
Training Ingredients						
GRPO	25.43	35.81	31.72	27.25	18.71	13.66
GRPO + Entropy	24.23	33.53	29.42	27.66	16.67	13.85
GRPO + Strategy-Conditioned Grouping	27.80	43.93	35.55	29.60	16.33	13.57
GRPO + Strategy-Conditioned Grouping + Entropy	**29.59**	47.00	38.22	31.58	15.99	15.18
Entropy Sensitivity (with Strategy-Conditioned Grouping)						
Entropy=0.01 (Reward Collapse)	—	—	—	—	—	—
Entropy=0.001	**29.59**	47.00	38.22	31.58	15.99	15.18
Entropy=0.0001	27.48	40.95	34.11	32.32	14.29	15.73

D Entropy Analysis

To separate ARC's core mechanism from setting-specific stabilizers, Table 6 fixes the reward family and varies only two ingredients: strategy-conditioned grouping and entropy regularization. The first block asks which component drives the main gain over GRPO; the second block studies entropy sensitivity once grouping is enabled.

Table 6 supports the paper's main causal story. Holding the reward design fixed, strategy-conditioned grouping is the primary source of improvement over base GRPO, whereas entropy regularization alone does not explain the gain and can even weaken performance. Entropy becomes useful only after the comparison class has been cleaned up by grouping, where it acts as a stabilizer for multi-channel generation rather than as the main mechanism. The entropy sweep further shows that this effect is sensitive to scale: too much entropy leads to reward collapse, while a moderate value of 0.001 gives the best overall operating point. Taken together, these results suggest that ARC helps mainly by changing *how* rollouts are compared, with entropy regularization serving as a secondary component that improves the stability of that mechanism in the full INTER^3 recipe.

E Annotation Guidelines for INTER^3-86K

This section provides the detailed annotation guidelines used by both models and human annotators for strategy assignment.

Progress Update: Proactively provide progress updates to the user when simultaneously executing a tool call or multi-step tasks. Notify the user of the current progress after completing each tool call/sub-task. Let the user know the task is in progress to reduce waiting anxiety.

Direct Answer: Answer directly without invoking any tools if the information is already known or contextually available. Be concise and clear; avoid verbosity.

Clarify First: If the user's request lacks sufficient information, ask for clarification first. Do not guess the user's intent; ask directly. Execute tool calls only after the user confirms. Suitable for vague, ambiguous, or incomplete requests.

Alignment Check: Restate the user's requirements first to confirm your understanding is correct. Execute only after the user confirms. Avoid making mistakes due to misunderstanding. Suitable for easily misunderstood, important, or irreversible operations.

F Mechanism-Level Analysis and Label-Noise Robustness

To complement the idealized theoretical analysis, we perform rollout-level analyses of empirical reward unfairness, realized inter- and intra-strategy reward variance, strategy adherence, and sensitivity to strategy-label noise.

Table 8: Sensitivity of ARC to strategy-label corruption. The corrupted setting replaces 50% of strategy instructions with a uniformly sampled alternative strategy during training. Results are based on a single training run (seed 1); values after ± denote standard deviations over three independent evaluation runs.

Method	Tool-use Avg.	τ-Airline	τ-Retail	τ^2-Airline	τ^2-Retail	τ^2-Telecom
GRPO	33.35	31.33 ± 4.62	40.29 ± 4.94	36.67 ± 6.43	40.64 ± 5.97	17.84 ± 2.82
GRPO + ARC	41.73	44.00 ± 3.06	50.00 ± 1.33	48.00 ± 6.11	45.61 ± 0.86	21.05 ± 0.91
GRPO + ARC, 50% corrupted	**35.01**	$\mathbf{34.00 \pm 3.46}$	$\mathbf{42.90 \pm 1.33}$	$\mathbf{37.16 \pm 0.73}$	$\mathbf{42.11 \pm 3.16}$	$\mathbf{18.87 \pm 1.07}$

Realized inter- and intra-strategy reward variance. On actual rollouts, we estimate the realized between-strategy and within-strategy reward variances. As shown in Table 7, the estimated inter-/intra-strategy variance ratio decreases from 0.417 under GRPO to 0.074 under ARC, corresponding to an approximately 82% reduction. This is consistent with the proposed variance-reduction mechanism. However, the confidence intervals are wide because relatively few prompt groups contain multiple realized strategies, so we interpret this result as mechanism-level evidence rather than definitive empirical validation of the stylized theoretical analysis.

Table 7: Estimated realized inter- and intra-strategy reward variance on rollout data. Ratio denotes $\hat{\sigma}^2_{\text{inter}}/\hat{\sigma}^2_{\text{intra}}$, with 95% confidence intervals.

Method	$\hat{\sigma}^2_{\text{inter}}$	$\hat{\sigma}^2_{\text{intra}}$	Ratio (95% CI)
GRPO	0.0555	0.1331	0.417 [0.002, 3.015]
ARC	0.0030	0.0401	0.074 [0.002, 0.410]

Sensitivity to strategy-label noise. We additionally test the sensitivity of ARC to imperfect strategy labels by corrupting 50% of the strategy instructions during training. For each corrupted example, the original strategy instruction is replaced with a uniformly sampled alternative strategy. This experiment uses one training seed, and the reported standard deviations are computed over three independent evaluations.

As shown in Table 8, ARC remains above the GRPO baseline under substantial label corruption, with a tool-use average of 35.01 compared with 33.35 for GRPO. However, performance is substantially lower than clean ARC at 41.73, indicating that ARC benefits from accurate strategy assignments and degrades under noisy training labels.

G Interaction Strategy Examples

G.1 Progress Update Strategy

```
[user]
Help me analyze Acme Corp's financial status.

[assistant]
The user wants a full financial analysis. I'll start by fetching the
    income statement.
<tool_call>
{"name": "get_income_statement", "arguments": {"company": "Acme Corp"}}
</tool_call>
<answer>I'm retrieving Acme Corp's financial statements - step 1 of 3...
    </answer>

[tool]
get_income_statement -> {"revenue": "$5.2B", "net_income": "$430M"}

[assistant]
Income statement retrieved. Now fetching the balance sheet for ratio
    analysis.
<tool_call>
{"name": "get_balance_sheet", "arguments": {"company": "Acme Corp"}}
```

```
</tool_call>
<answer>Income statement retrieved. Now analyzing the balance sheet -
    step 2 of 3...</answer>
```

G.2 Direct Answer Strategy

```
[user]
What is the capital of China?

[assistant]
<answer>The capital of China is Beijing.</answer>
```

G.3 Clarify First Strategy

```
[user]
Book a meeting room for me.

[assistant]
Request lacks required information: time and capacity.
<answer>I'd be happy to book a meeting room. Could you tell me:
(1) What date and time?
(2) How many people will attend?</answer>

[user]
Tomorrow at 10 AM, 8 people.

[assistant]
Details confirmed. Proceeding with booking.
<tool_call>{"name": "book_meeting_room", "arguments": {"date":
    "tomorrow", "time": "10:00", "capacity": 8}}</tool_call>
<answer>Proceeding with booking a room for tomorrow at 10 AM suitable
    for 8 people...</answer>

[tool]
book_meeting_room -> {"status": "confirmed", "room": "Room 3B"}

[assistant]
<answer>Booked! Room 3B for tomorrow at 10:00 AM, capacity 8.</answer>
```

G.4 Alignment Check Strategy

```
[user]
Send last quarter's sales data to Manager Wang.

[assistant]
This is a send operation --- I should confirm the exact scope and
    recipient before executing.
<answer>Just to confirm: you'd like me to send the Q1 2024
    (January-March) sales data report to Manager Wang at
    wang@company.com. Is that correct?</answer>

[user]
Yes, that's right.

[assistant]
Intent confirmed. Executing send operation.
<tool_call>{"name": "send_email", "arguments": {"to":
    "wang@company.com", "subject": "Q1 2024 Sales Data", "attachment":
    "q1_sales.pdf"}}</tool_call>

[tool]
send_email -> {"status": "sent"}
```

```
[assistant]
<answer>Sent! Q1 2024 sales report delivered to Manager Wang.</answer>
```

H System Prompt

The system prompt used for training and evaluation:

System Prompt Configuration

System Role

You are an advanced AI assistant capable of streamingly interleaved interactions, reasoning, and tool executions. Do not wait for the entire process to finish. You can provide partial updates in `<answer>` while continuing to reason and call tools.

Output Requirements

Your output must be structured into functional segments. Texts that are NOT wrapped in `<answer>` tags are treated as internal reasoning and are hidden from the user. Ensure all XML-style tags (`<tool_call>`, `<answer>`) are strictly paired and properly closed.

Tag Definitions

- **Internal Reasoning:** (No tag required) Use plain text outside of any tags for logical reasoning, task decomposition, or analyzing tool outputs. This is hidden from the user.
- **`<answer>`:** (User Visible) This is the ONLY content displayed to the user. Use this for status updates, partial answers, or the final conclusion. If your response relies on information from the tool outputs, do not expose the raw tool output verbatim. Instead, integrate and summarize the relevant facts naturally in your own words.
- **`<tool_call>`:** (Internal Only) Use this to call external functions using the provided tools, if any.

I Evaluation Setup and Configuration

This appendix describes the standardized evaluation configuration used across all benchmarks reported in Table 2.

I.1 Inference Configuration

All evaluations were conducted using the following controlled parameters:

Concurrency. TTFT measurements are collected with concurrency level set to 16 parallel requests on the same GPU (H200). This concurrency level is maintained consistently across all τ-bench evaluations to ensure fair latency comparisons.

Interleaved Mode. When interleaved mode is enabled, we use the same structured interface as in training: `<answer>` for user-visible content and `<tool_call>` for tool calls. This enables fine-grained measurement of time-to-first-token (TTFT) as the model can provide partial answers while continuing reasoning and tool execution.

I.2 Benchmark Configurations

τ-Bench. Following the official τ-Bench evaluation protocol (Yao et al., 2025), we evaluate tool-calling ability across the *retail* and *airline* domains. The user simulator is powered by the Qwen3-235B-A22B-Instruct model (Qwen Team, 2025). We report the average reward score (success rate) and TTFT. Each evaluation run is repeated 3 times and we report the mean score.

τ^2-Bench. Following the official τ^2-Bench evaluation protocol (Barres et al., 2025), we evaluate on the *retail*, *airline*, and *telecom* domains. The user simulator is powered by the Qwen3-235B-A22B-Instruct model (Qwen Team, 2025). We repeat the evaluation 3 times and we report the mean score.

AIME 2026. The American Invitational Mathematics Examination 2026 dataset (Balunovic et al., 2025) contains challenging math problems with integer answers in [0, 999]. We extract the model's answer from `\boxed{}` notation. Each problem is evaluated 8 times with different random seeds, and accuracy is computed by exact match after normalization. The instruction prompt asks the model to solve the problem and format the final answer.

GPQA Diamond. The Google-Proof Q&A (GPQA) Diamond subset (Rein et al., 2024) contains expert-level science questions with multiple-choice answers. We evaluate the model's ability to select the correct option (A, B, C, or D). Each question is run 8 times, and we compute the accuracy as the fraction of correct selections.

HMMT 2025. The Harvard-MIT Mathematics Tournament February 2025 dataset (Balunovic et al., 2025) contains competition-level math problems across multiple rounds. Answers may be integers, fractions, expressions, or other mathematical objects. Each problem is evaluated 8 times.

IFBench. Instruction Following Benchmark (Pyatkin et al., 2025) tests the model's ability to follow complex, multi-constraint instructions. The benchmark contains test cases with specific formatting, length, and content requirements. Evaluation is performed using the official IFBench scoring script, which checks constraint satisfaction.

Arena-Hard. Following the official Arena-Hard evaluation protocol (Li et al., 2025), we use the benchmark's GPT-4-based judge to compare model responses against a baseline (o3-mini-2025-01-31) on 580 challenging questions from diverse domains. We report the win rate percentage as the final score.

I.3 TTFT Measurement Protocol

Time-to-first-token (TTFT) is measured specifically for tool-calling benchmarks (τ-bench variants) where latency is most critical. The measurement protocol:

- **Definition:** TTFT is the time from sending the request to receiving the first **user-visible** token of the model's response.
- **Aggregation:** For each row in Table 2, we report the mean TTFT over all available τ-bench domain measurements for that configuration.
- **Conditions:** All TTFT measurements use concurrency=16 on H200 GPUs, with no other concurrent workloads.
- **Scope:** TTFT is not reported for reasoning benchmarks (AIME, GPQA, HMMT) or instruction-following/alignment benchmarks (IFBench, Arena-Hard) as these are primarily accuracy-focused rather than latency-sensitive.

I.4 Evaluation Uncertainty and Training-Seed Robustness

We additionally evaluate the statistical variability of ARC under both repeated evaluation and independent training seeds. These analyses separate variability arising from stochastic evaluation from variability arising from optimization randomness during training.

Evaluation uncertainty. Table 9 reports variability across repeated evaluations of the same trained checkpoint. For the main τ-bench and τ^2-bench results, we perform three independent evaluation runs for each model checkpoint and report the mean and standard deviation across runs. In particular, ARC increases the tool-use average from 33.60 to 38.45 for PPO and from 33.35 to 41.73 for GRPO. The corresponding DAPO results are substantially closer, indicating that the effect of ARC depends on the underlying RL backbone.

Table 9: Evaluation uncertainty on the tool-use benchmarks. Results are reported as mean ± standard deviation over $N = 3$ independent evaluation runs using the same trained checkpoint. Tool-use Avg. is the mean over the five displayed τ-bench and τ^2-bench metrics.

Method	Tool-use Avg.	τ-Airline	τ-Retail	τ^2-Airline	τ^2-Retail	τ^2-Telecom
Qwen3-8B-NoThink	22.16 ± 2.37	12.00 ± 0.00	29.86 ± 5.31	14.61 ± 7.79	36.55 ± 1.01	17.80 ± 2.53
Qwen3-8B-Think	31.35 ± 0.19	28.00 ± 6.00	36.81 ± 4.29	29.75 ± 3.27	38.71 ± 1.06	23.46 ± 4.19
PPO	33.60 ± 1.34	35.33 ± 3.06	41.45 ± 3.92	33.33 ± 3.06	38.89 ± 2.21	19.01 ± 1.34
PPO + ARC	$\mathbf{38.45 \pm 0.40}$	$\mathbf{39.33 \pm 2.31}$	$\mathbf{46.09 \pm 1.74}$	$\mathbf{41.61 \pm 0.68}$	$\mathbf{44.44 \pm 3.65}$	$\mathbf{20.76 \pm 2.21}$
DAPO	34.09 ± 3.26	31.33 ± 4.16	42.32 ± 5.52	35.33 ± 7.57	42.11 ± 4.88	19.37 ± 3.40
DAPO + ARC	$\mathbf{34.24 \pm 1.89}$	$\mathbf{34.67 \pm 2.31}$	$\mathbf{41.74 \pm 3.98}$	$\mathbf{34.50 \pm 3.97}$	$\mathbf{40.64 \pm 7.04}$	$\mathbf{19.64 \pm 2.79}$
GRPO	33.35 ± 3.04	31.33 ± 4.62	40.29 ± 4.94	36.67 ± 6.43	40.64 ± 5.97	17.84 ± 2.82
GRPO + ARC	$\mathbf{41.73 \pm 2.11}$	$\mathbf{44.00 \pm 3.06}$	$\mathbf{50.00 \pm 1.33}$	$\mathbf{48.00 \pm 6.11}$	$\mathbf{45.61 \pm 0.86}$	$\mathbf{21.05 \pm 0.91}$

Table 10: Robustness to training randomness. Results are reported as mean ± standard deviation over three independently trained models with different training seeds. Each seed-level score is obtained by averaging repeated evaluation runs for that checkpoint.

Method	Tool-use Avg.	τ-Airline	τ-Retail	τ^2-Airline	τ^2-Retail	τ^2-Telecom
PPO	31.81 ± 1.87	33.78 ± 3.29	36.13 ± 5.09	32.89 ± 3.36	37.14 ± 1.92	19.10 ± 0.17
PPO + ARC	$\mathbf{36.76 \pm 1.56}$	$\mathbf{37.78 \pm 1.39}$	$\mathbf{44.83 \pm 2.96}$	$\mathbf{37.28 \pm 3.78}$	$\mathbf{43.84 \pm 0.59}$	$\mathbf{20.08 \pm 1.18}$
DAPO	34.69 ± 0.52	35.55 ± 4.02	40.97 ± 1.31	35.56 ± 1.02	41.91 ± 0.34	19.44 ± 0.46
DAPO + ARC	$\mathbf{34.77 \pm 0.70}$	$\mathbf{32.89 \pm 1.68}$	$\mathbf{42.22 \pm 3.36}$	$\mathbf{37.50 \pm 2.62}$	$\mathbf{41.13 \pm 0.85}$	$\mathbf{20.11 \pm 0.58}$
GRPO	34.74 ± 1.35	32.67 ± 1.34	42.61 ± 2.03	38.20 ± 1.37	41.80 ± 3.36	18.44 ± 0.80
GRPO + ARC	$\mathbf{37.53 \pm 1.53}$	$\mathbf{39.11 \pm 1.68}$	$\mathbf{43.87 \pm 4.36}$	$\mathbf{40.19 \pm 1.65}$	$\mathbf{43.84 \pm 1.60}$	$\mathbf{20.67 \pm 1.53}$

Robustness to training randomness. Table 10 evaluates sensitivity to optimization randomness across three independent training seeds. Each resulting checkpoint is evaluated using the same repeated-evaluation protocol. We first average the repeated evaluations within each seed and then report the mean and standard deviation across the three seed-level scores.

The improvements are consistent across training seeds for PPO and GRPO. PPO improves from 31.81 to 36.76 on the tool-use average, while GRPO improves from 34.74 to 37.53. In contrast, DAPO remains essentially unchanged (34.69 versus 34.77), and we therefore do not interpret the DAPO result as evidence of a meaningful tool-use improvement.

J Training Details

J.1 Hyperparameters

Table 11 details RL training hyperparameters.

Table 11: RL hyperparameters for the main experiments.

Parameter	Value
Base Model	Qwen3-8B
Learning Rate	1e-6
Batch Size	64
PPO Mini Batch Size	8
PPO Micro Batch Size (per GPU)	4
Max Prompt Length	6000
Max Response Length	1024
Training Epochs	1
KL Loss Coefficient	0.001
KL Loss Type	low_var_kl
Use KL in Reward	False
GRPO Groups (n)	8
Entropy Coefficient	0.001



Figure 6: Training-time instruction removal underperforms the default ARC setup. Left: average number of off-strategy rollouts per group, where an rollout has a predicted strategy different from the injected strategy. Right: standard deviation of off-strategy counts across groups. Lower values are better in both panels, indicating more consistent within-strategy rollout grouping.

Table 12: Curriculum learning over strategy-instruction removal does not improve ARC: no removal performs best, linear removal is intermediate, and constant removal is weakest.

Training Setting	**Avg.**	**τ Avg.**	**τ^2 Avg.**	**Reasoning Avg.**	**IFBench**	**ArenaHard**
No removal (ARC default)	29.59	47.00	38.22	31.58	15.99	15.18
Linear removal	27.28	38.21	31.83	34.12	16.33	15.92
Constant removal	26.97	34.97	32.99	33.21	18.71	15.01

J.2 Compute Resources

All experiments were conducted on NVIDIA H200 GPUs, using NVIDIA driver version 570.158.01 and CUDA 13.0.

K Additional Experimental Analyses

K.1 Curriculum Learning over Training-Time Strategy Instructions

ARC uses strategy instructions during training but removes them at inference, motivating a curriculum-learning question: should the conditioning signal be gradually weakened so the policy learns more autonomous strategy selection without losing ARC's variance-control benefit. Inspired by curriculum-learning methods in RL (Bengio et al., 2009; Florensa et al., 2018b;a; Matiisen et al., 2017), we study this as a diagnostic analysis of the trade-off between train-inference alignment and within-strategy comparability.

We compare three settings: (1) *No removal*, the default ARC setup that always retains the strategy instruction; (2) *Linear removal*, where the instruction-drop probability increases over training according to Appendix A, $p_d(t) = p_{\min} + (p_{\max} - p_{\min})\frac{t}{T}$; and (3) *Constant removal*, which drops instructions with fixed probability $p_d = 0.20$ throughout training.

Table 12 and Figure 6 show that progressively removing strategy instructions does not improve ARC in our setting. Default ARC achieves the strongest τ/τ^2 tool-use performance and the lowest off-strategy mean and variance, indicating more stable within-strategy rollout grouping. Linear removal provides a partial trade-off by modestly improving some reasoning metrics at the cost of agentic performance, while constant removal performs worst overall. These results suggest that strategy instructions are most effective as a persistent training-time variance-control mechanism rather than a signal to be annealed away. This is consistent with ARC's mechanism: strategy instructions are most useful as a training-time variance-control device, and annealing them away too early weakens within-strategy comparability instead of improving the final policy.

Table 13: Strategy scaling ablation. Incremental strategy scaling consistently drives overall performance gains: the full strategy suite performs best, while reasoning tasks peak with three strategies and agentic benchmarks show the most dramatic sensitivity to strategy diversity.

Strategy Segment	Avg.	τ Avg.	τ^2 Avg.	Reasoning Avg.	IFBench	ArenaHard
Progress Update	21.74	23.62	22.36	34.01	16.33	14.27
Progress Update + Direct Answer	24.91	31.67	32.46	27.79	17.69	14.96
Progress Update + Direct Answer + Clarify First	25.60	31.04	28.97	35.21	16.67	16.13
Full Suite (4)	29.59	47.00	38.22	31.58	15.99	15.18

K.2 Strategy Scalability Analysis

We conduct an additional ablation study of strategy scaling in Table 13, which reveals a clear performance trajectory as strategy components are incrementally integrated, moving from the highest to the lowest training data volume. The *Avg.* shows a consistent upward trend, achieving an impressive 36.1% total improvement. The most significant gains are concentrated in agentic benchmarks, where performance on τ-bench and τ^2-bench surged by 99% and 71%, respectively, as the model transitioned from a single strategy to the full suite.

These results highlight not only the effectiveness of the method in leveraging heterogeneous strategies, but also its scalability: as additional strategies are added, the model continues to improve, demonstrating that even minority strategies—such as *Alignment Check*, which constitutes only 2.5% of the data—contribute meaningfully to overall capabilities. This highlights the potential for accommodating larger strategy sets without performance degradation.

K.3 Additional Diagnostics on Strategy Instructions

We evaluate whether the final ARC policy depends on explicit strategy hints at deployment, or whether the strategy behavior has already been internalized during training. We compare matched, randomized, and removed instructions on the same ARC checkpoint across all τ-bench and τ^2-bench submetrics to isolate the effect of prompt-side cues.

Conclusively, Figure 7 shows that inference-time strategy hints do not improve the deployed policy. The baseline with no strategy hint removal managed to achieve the highest overall average (41.73), outperforming both Randomized Instruction (39.58) and Matched Instruction (39.85). Domain-level analysis reveals that matched instructions help on τ Airline but hurt on τ^2 Airline, while all settings struggle on τ^2 Telecom. This domain-dependent variability confirms that inference-time hints lack a reliable deployment benefit. Strategy instructions serve their primary function during training by structuring comparison groups, not as persistent cues that should remain at inference.

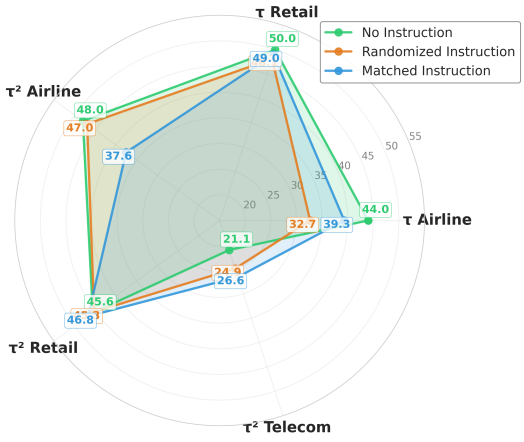

Figure 7: Domain-instruction interaction analysis. Inference-time strategy hints show no uniformly superior deployment interface.