PARETO-GRPO: PROMPT-LOCAL PARETO CREDIT ASSIGNMENT FOR EFFICIENT SEARCH-AUGMENTED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-relative reinforcement learning works well when a verifier supplies a clear scalar reward, but search agents expose a missing interface: trajectories can reach the same answer while using very different amounts of search. Fixed scalarizations resolve such ties by imposing one global exchange rate between correctness and cost, even though the useful amount of search is prompt dependent. We introduce Pareto-GRPO, which keeps the critic-free GRPO update but orders each prompt's rollout group by Pareto dominance over correctness and search efficiency. A bounded rank utility then refines the group-normalized correctness advantage. This ordinal construction is invariant to coordinate-wise monotone reward transforms and admits an exact first-order objective characterization. In a controlled Qwen2.5-7B comparison with identical initialization, data, retrieval, optimization, and validation-only model selection, Pareto-GRPO reaches 0.465 average exact match with 1.78 searches and 0.286 hypervolume. The strongest tuned scalar baseline reaches 0.441, 1.85, and 0.258, respectively; all five seed-wise accuracy differences favor Pareto-GRPO. The advantage persists under enforced search caps, alternative cost definitions, and a three-objective extension. Across seven benchmarks and two model scales, Pareto credit assignment improves accuracy, efficiency, and run-to-run stability together.

1 INTRODUCTION

Search-augmented language models decide not only what to answer but also whether to query a tool, how often to query it, and when to stop (Yao et al., 2023; Schick et al., 2023; Nakano et al., 2021; Asai et al., 2023). Reinforcement learning can improve these decisions, yet search-agent training remains unusually brittle: additional calls may uncover decisive evidence, repeat information already found, or inject retrieval noise (Jin et al., 2025b;a; Sun et al., 2025; Trivedi et al., 2023). A final answer reward cannot tell these cases apart.

This ambiguity is acute in Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO normalizes rewards among rollouts sampled for the same prompt, but a binary verifier makes every correct–correct and wrong–wrong pair a tie. A weighted correctness–cost reward adds information, although it also assumes that one cardinal exchange rate is appropriate for an easy factoid and a difficult multi-hop question. Prompt-local scalarization and decoupled multi-reward normalization are stronger alternatives (Roijers et al., 2013; Hayes et al., 2022; Liu et al., 2026); neither removes that cardinal choice.

We instead use the structure already present in a GRPO batch. Pareto-GRPO compares trajectories only within the same prompt and asks whether one is no worse in correctness and efficiency and strictly better in at least one. Non-dominated sorting turns these comparisons into ranks, and a bounded rank utility augments the original correctness advantage. The method therefore supplies an *ordinal reward-vector-to-advantage interface*: it uses objective values to determine dominance, not to define a global conversion between accuracy and cost. Figure 1 shows the mechanism.

The distinction is empirical, not cosmetic. Under one controlled protocol, tuned scalar penalty, weighted-sum, prompt-local scalarization, and GDPO-style baselines reach 0.405, 0.423, 0.441, and 0.438 exact match; Pareto-GRPO reaches 0.465 while also using fewer searches. Removing or

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.

shuffling the ranks erases most of this gap. Gains remain under hard inference caps, when search count is replaced by token or latency cost, and with three objectives. Thus the result is not an under-tuned scalar baseline, a favorable realized-budget slice, or a peculiarity of one efficiency transform.

Our contributions are:

- We identify prompt-local ordinal credit assignment as the missing link between multi-reward feedback and GRPO, and instantiate it with non-dominated sorting over trajectory groups.
- We characterize the expected update as the gradient of a fixed-behavior first-order objective. We also establish dominance consistency, invariance to separate strictly increasing coordinate transforms, bounded shaping, and a sufficient condition that shaping cannot reverse a correct–incorrect preference.
- We provide a controlled seven-benchmark study at two model scales. The study includes full baseline sweeps, five-seed paired comparisons, source-isolation ablations, hard caps, alternative costs, and a three-objective extension.

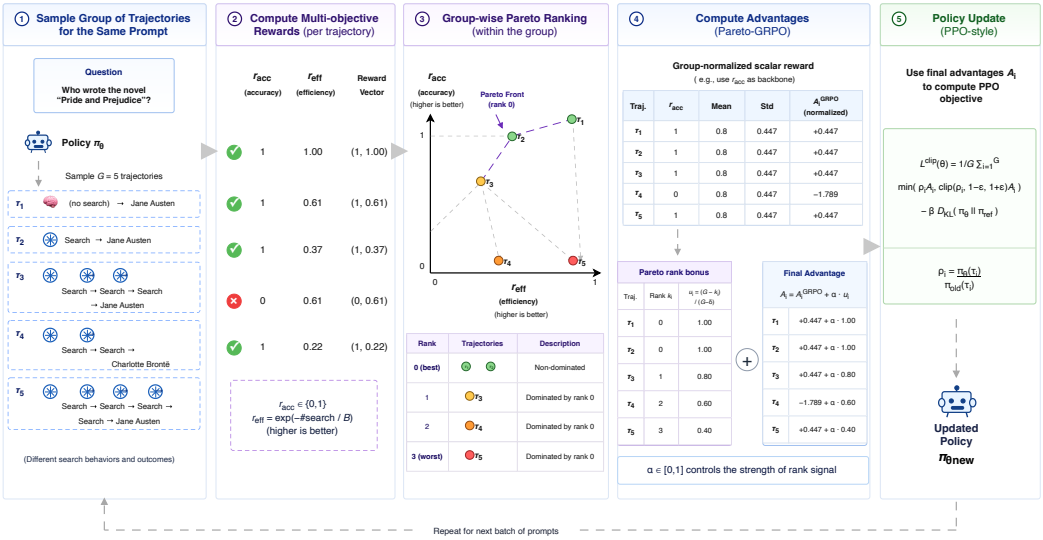

Figure 1: **Prompt-local Pareto credit assignment.** Outcome-only GRPO ties trajectories with the same answer reward. Pareto-GRPO forms Pareto fronts over correctness and efficiency within each prompt group, then uses rank utility to separate otherwise tied trajectories. The panels illustrate the ordering rule; observed rollout examples appear in Figure 5.

2 Related Work

Search agents and group-relative RL. Tool-using models interleave generation with retrieval, browsing, or execution (Yao et al., 2023; Schick et al., 2023; Nakano et al., 2021; Gao et al., 2023; Shinn et al., 2023; Asai et al., 2023). Search-R1 and follow-up studies train these decisions with policy optimization and document sensitivity to reward design and retrieval noise (Jin et al., 2025b;a; Sun et al., 2025). GRPO is attractive here because it avoids a learned critic, but its group normalization cannot order rollouts that share a binary outcome (Shao et al., 2024). We keep its policy update and change the construction of group advantages.

Multi-reward optimization. Multi-objective RL studies scalarization, constrained optimization, and policy sets that cover a Pareto frontier (Roijers et al., 2013; Hayes et al., 2022). GDPO instead decouples normalization across multiple rewards for language-model RL (Liu et al., 2026). Our aim differs from learning a preference-conditioned policy set: we use non-dominated sorting (Deb et al., 2002) as a prompt-local credit assignment operator. This makes the comparison ordinal and

leaves correctness as the scalar backbone. The controlled study compares against fixed, prompt-local, and decoupled-normalization alternatives rather than relying on outcome-only GRPO as the sole reference.

Credit assignment. Delayed outcomes create high-variance supervision (Arjona-Medina et al., 2019), while reasoning and search admit many trajectories with the same endpoint (Wei et al., 2022; Wang et al., 2023; Trivedi et al., 2023). Process rewards add denser supervision when step labels or learned judges are available. Pareto-GRPO addresses a complementary case: the environment already returns several trajectory-level objectives, and the unresolved question is how to convert their partial order into a stable group-relative advantage.

3 METHOD

3.1 FROM REWARD VECTORS TO GROUP ADVANTAGES

For prompt x, policy π_θ samples G trajectories $\mathcal{T}_x = \{\tau_i\}_{i=1}^G$. Each trajectory receives $\mathbf{r}_i = [r_{\text{acc}}(\tau_i), r_{\text{eff}}(\tau_i)]$, where

$$r_{\text{acc}}(\tau_i) = \mathbb{1}[\text{answer is correct}], \tag{1}$$

$$r_{\text{eff}}(\tau_i) = \exp\left(-\frac{n_{\text{search}}(\tau_i)}{B}\right), \tag{2}$$

and B is the search budget. Here n_{search} is the rollout logger's count of parsed, well-formed search actions; Appendix D gives the exact accounting rule. Correctness remains the scalar GRPO reward. Efficiency is used to order trajectories rather than added to correctness with a fixed weight.

3.2 GROUP-WISE PARETO RANKING

Within a prompt group, τ_a dominates τ_b, written $\tau_a \succ \tau_b$, if

$$r_{\text{acc}}(\tau_a) \geq r_{\text{acc}}(\tau_b), \qquad r_{\text{eff}}(\tau_a) \geq r_{\text{eff}}(\tau_b),$$

with at least one strict inequality. Repeatedly removing the non-dominated set produces fronts $F^{(0)}, F^{(1)}, \ldots$; $k_i = m$ when $\tau_i \in F^{(m)}$. Lower rank is better. The advantage is

$$A_i = z_i + \alpha u_i, \qquad z_i = \frac{r_{\text{acc},i} - \mu_x}{\sigma_x + \epsilon}, \quad u_i = \frac{G - k_i}{G}. \tag{3}$$

Thus $u_i \in [1/G, 1]$ refines the standard correctness advantage without replacing it. A cheap wrong trajectory may be non-dominated when the group contains no correct trajectory that is at least as efficient, but the bounded term cannot overwhelm an arbitrarily chosen correctness margin; Proposition 4 gives a sufficient condition. We use the uncentered form above and report a centered variant in Appendix L.

3.3 WHAT OBJECTIVE DOES THE UPDATE FOLLOW?

The rank depends on the other sampled trajectories, so Eq. 3 should not be interpreted as a global, clipped hypervolume objective. It nevertheless has an exact local characterization. Fix a behavior policy $\bar{\pi}$ and define

$$\bar{A}_G^{\bar{\pi}}(\tau, x) = \mathbb{E}_{\tau_{2:G} \sim \bar{\pi}}\left[z(\tau, \tau_{2:G}) + \alpha u(\tau, \tau_{2:G})\right], \tag{4}$$

$$J_G(\theta; \bar{\pi}) = \mathbb{E}_x \mathbb{E}_{\tau \sim \pi_\theta(\cdot|x)}[\bar{A}_G^{\bar{\pi}}(\tau, x)]. \tag{5}$$

At $\theta = \bar{\theta}$, the score-function gradient of J_G equals the expected group-average Pareto-GRPO estimator (Proposition 1). This states precisely what the unclipped on-policy update follows to first order. It does not claim monotonic improvement of a global hypervolume after finite clipped steps.

Because ranks depend only on comparisons, separate strictly increasing transforms of any objective leave the dominance graph, fronts, and update unchanged. This explains why $\exp(-n/B)$, $1/(1+n)$, and $1 - n/B$ produce identical rankings when applied to the same search count. The appendix proves this invariance, dominance consistency, boundedness, and order preservation.

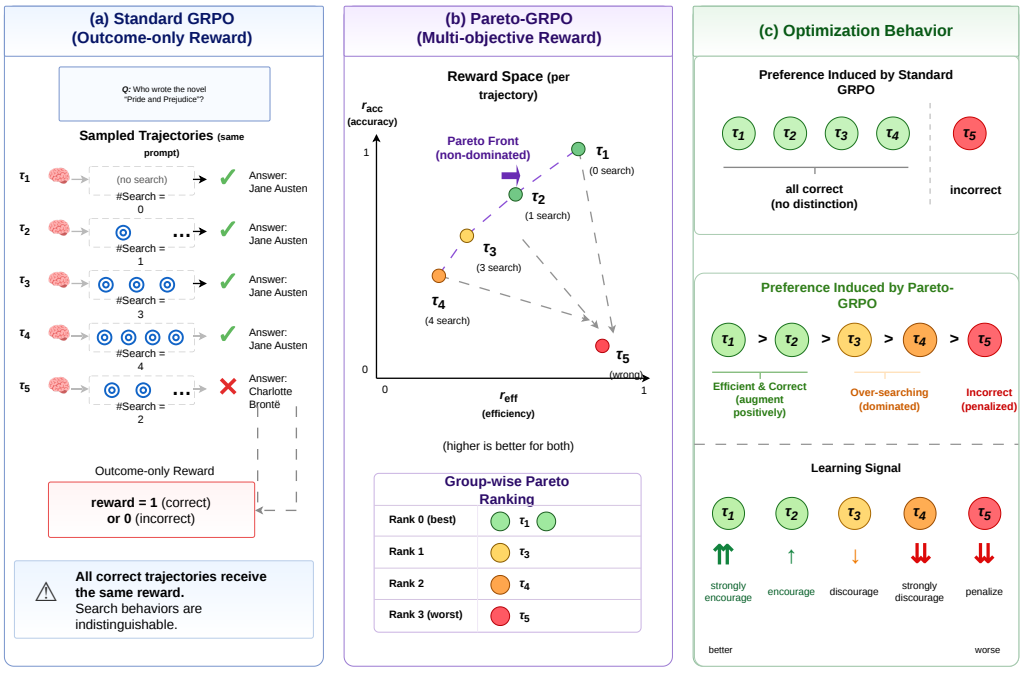

Figure 2: **Training pipeline.** Sample G trajectories per prompt, compute reward vectors, obtain non-dominated ranks, and add bounded rank utility to the scalar GRPO advantage before the usual clipped policy update.

4 Experiments

Protocol. We train from the same Qwen2.5-7B initialization on NQ and HotpotQA and evaluate on seven single- and multi-hop benchmarks. Every controlled method uses the same retrieval index, prompts, group size $G = 5$, search budget $B = 4$, decoding, optimizer, KL coefficient, steps, evaluator, and five seeds. Hyperparameters are selected on validation hypervolume only; reported rows are the resulting seven-dataset test/dev measurements. Baseline sweeps consume roughly 1,000–1,200 GPU-hours per family and Pareto-GRPO about 800 GPU-hours including final runs, with the same early-stopping rule. Appendix D records the grids and implementation details. Hypervolume (HV) uses normalized coordinates $(\mathrm{EM}, 1 - \min(\bar{n}_{\mathrm{search}}, B)/B)$ and reference point $(0, 0)$.

4.1 Generalization across datasets and model scales

Table 1 reports the path-matched outcome-only control and Pareto-GRPO on seven benchmarks. The 7B average rises from 0.350 to 0.465, and the 3B average from 0.312 to 0.402. Every dataset improves over outcome-only GRPO. The largest 7B gains occur on HotpotQA and 2Wiki ($+0.146$ each), where a policy must decide how much evidence to gather rather than merely whether to retrieve once.

4.2 Controlled comparison against tuned multi-reward baselines

Table 2 is the central comparison. Each baseline receives a full validation sweep: scalar penalty $\lambda \in \{0, .02, .05, .10, .20, .30\}$; weighted sum $\lambda \in \{0, .02, .05, .10, .20, .30, .50\}$; prompt-local scalarization $\lambda \in \{.02, .05, .10, .20, .30, .50\}$; GDPO-style normalization $\lambda \in \{.05, .10, .20, .50, 1.0\}$; and Pareto-GRPO $\alpha \in \{.05, .10, .20\}$. The selected values are .10, .20, .10, .20, and .10, respectively. These are not validation scores: after validation-only selection, every row is evaluated on the same seven-dataset test/dev suite.

Against prompt-local scalarization, the strongest controlled baseline, Pareto-GRPO gains 0.024 EM (paired bootstrap 95% CI $[0.014, 0.034]$), reduces search by 0.070, and gains 0.028 HV. All five

Table 1: **Exact match across search-QA benchmarks.** Both rows at each scale use the same initialization and pipeline. Avg. is the unweighted mean.

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-7B								
Outcome-only GRPO	.395	.560	.388	.326	.297	.125	.360	.350
Pareto-GRPO	**.506**	**.673**	**.479**	**.472**	**.443**	**.225**	**.460**	**.465**
Qwen2.5-3B								
Outcome-only GRPO	.421	.583	.413	.297	.274	.066	.128	.312
Pareto-GRPO	**.477**	**.637**	**.463**	**.372**	**.387**	**.143**	**.334**	**.402**

Table 2: **Controlled Qwen2.5-7B comparison.** Five seeds; identical initialization, data, retrieval, rollout, optimization, and evaluator. ↓ is better for search and seed standard deviation.

Method	EM ↑	Search ↓	HV ↑	Seed std. ↓
Outcome-only GRPO	.350	2.680	.176	.061
Scalar search penalty	.405	1.980	.223	.035
Weighted sum	.423	1.910	.241	.028
Prompt-local scalarization	.441	1.850	.258	.022
GDPO-style (Liu et al., 2026)	.438	1.880	.255	.020
Pareto-GRPO	**.465**	**1.780**	**.286**	**.010**

seed-wise EM differences are positive. It wins on six of seven datasets; the four multi-hop tasks improve by 0.032 macro-average. The sole non-win is PopQA at -0.005 (CI $[-0.018, 0.008]$). Against GDPO-style normalization, the corresponding differences are $+0.027$ EM, -0.100 searches, and $+0.031$ HV. Appendix E reports sweep endpoints and compute parity.

Table 3 makes the tuning check explicit. Increasing the cost weight beyond the selected value can reduce search for every scalarized baseline, but only by moving to a substantially worse accuracy–cost point. Pareto-GRPO remains stable over its smaller rank-weight grid and has the best selected hypervolume. This rules out the explanation that the comparison favors Pareto ranking simply because the scalar baselines were assigned weak cost penalties.

Table 3: **Validation-selected setting and cost-aggressive endpoint for each family.** Triplets are (EM/search/HV) from the common evaluation; they are not validation metrics.

Family	Selected setting	Cost-aggressive endpoint
Scalar penalty	.10: .405/1.980/.223	.30: .374/1.490/.211
Weighted sum	.20: .423/1.910/.241	.50: .389/1.310/.214
Prompt-local	.10: .441/1.850/.258	.50: .371/1.100/.204
GDPO-style	.20: .438/1.880/.255	1.0: .401/1.400/.226
Pareto-GRPO	.10: **.465/1.780/.286**	.20: .458/1.710/.279

Why prompt-local scalarization is not the same comparison. Prompt-local scalarization is a demanding baseline because it avoids applying one dataset-wide cost weight blindly. It still maps each reward vector to a number before the group is compared. Two consequences remain. First, a change in reward units changes the induced margins unless the weight is retuned, whereas a coordinate-wise monotone transform leaves Pareto ranks unchanged. Second, the scalar margin between two rollouts grows with reward magnitude even when only their order is reliable. Pareto-GRPO separates these roles: correctness supplies the cardinal GRPO backbone, while the auxiliary objectives contribute only dominance information. This is why the method can beat prompt-local scalarization without claiming that scalarization is intrinsically weak.

The comparison with weighted sum + rank is especially useful here. Both variants see exactly the same reward coordinates and the same tuned scalar signal; the only new information is the non-dominated rank. Its $+0.031$ EM and $+0.036$ HV increments therefore cannot be attributed to a different cost definition or a larger search penalty. Conversely, rank-only training is weaker than

the full method, confirming that ordinal shaping complements rather than replaces the correctness reward.

4.3 THE ORDERING SIGNAL CAUSES THE GAIN

Table 4 separates reward content from ordering. Efficiency-only optimization finds cheap failures. Shuffling ranks is statistically indistinguishable from outcome-only GRPO ($+0.006$ EM, CI $[-0.004, 0.016]$), so an extra random bonus is not enough. In contrast, adding the true rank to the best weighted sum improves EM by 0.031 (CI $[0.020, 0.042]$) and HV by 0.036. Full Pareto-GRPO adds the correctness backbone to rank-only training and gains another 0.033 EM. These interventions locate the improvement in the prompt-local dominance order.

Table 4: **Source isolation on Qwen2.5-7B.** "+ rank" adds the true Pareto utility to the best tuned weighted sum.

Training signal	EM ↑	Search ↓	HV ↑
Outcome-only	.350	2.680	.176
Efficiency-only	.285	**.880**	.197
Shuffled Pareto rank	.356	2.600	.182
Pareto rank only	.432	1.720	.257
Best weighted sum	.423	1.910	.241
Weighted sum + rank	.454	1.750	.277
Full Pareto-GRPO	**.465**	1.780	**.286**

The training curves tell the same story. Outcome-only GRPO has 0.061 seed standard deviation, a 0.271 worst seed, and two collapsed runs out of five; Pareto-GRPO reduces these values to 0.010, 0.454, and zero. Figure 3 shows that the improvement comes from smoother, repeatable optimization rather than a single lucky endpoint.

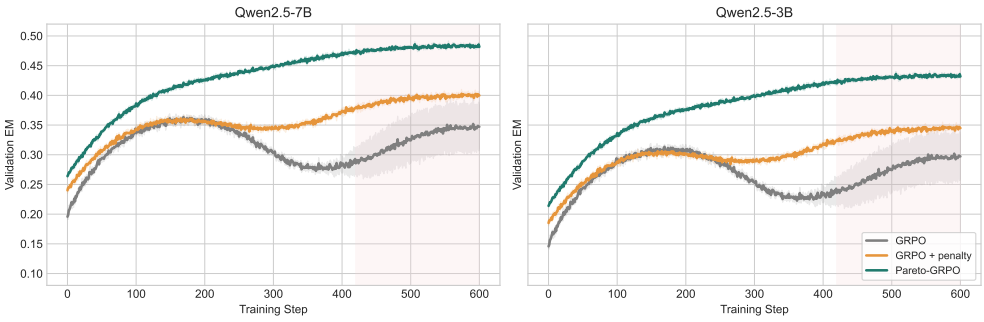

Figure 3: **Training stability.** Mean and standard deviation across seeds. Pareto-GRPO reaches a higher endpoint with substantially narrower dispersion.

The stability improvement holds at both scales (Table 5). On 7B, the search count on correct examples drops from 2.44 to 1.46, so the accuracy gain is not purchased by longer successful trajectories. On 3B, Pareto-GRPO similarly eliminates collapsed runs and lifts the worst seed by 0.082.

4.4 HARD CONSTRAINTS AND BROADER OBJECTIVES

Realized-budget conditioning can be confounded by policy selection, so we additionally enforce the inference cap. Pareto-GRPO wins at every cap in Table 6; at cap 2, the 0.025 gain has CI $[0.012, 0.038]$. The method therefore improves answers when both policies receive exactly the same maximum number of searches.

Realized allocation. Hard caps answer whether the advantage survives an equal resource ceiling; realized trajectories answer a different question—whether the learned policy allocates that resource selectively. For the 7B model, conditioning on logged usage gives the pattern in Table 7. These values

Table 5: **Efficiency and stability across scales.** Correct search is the average count on correctly answered examples.

Backbone	Method	Search	Correct search	Seed std.	Worst seed	Collapse
7B	Outcome-only	2.680	2.440	.061	.271	2/5
7B	Pareto-GRPO	**1.780**	**1.460**	**.010**	**.454**	**0/5**
3B	Outcome-only	2.570	2.310	.052	.247	2/5
3B	Pareto-GRPO	**1.960**	**1.700**	**.024**	**.329**	**0/5**

Table 6: **Exact match under enforced search caps.** The baseline is the strongest controlled method, prompt-local scalarization.

Hard cap	Strongest baseline	Pareto-GRPO	Difference
1	.330	**.345**	+.015
2	.405	**.430**	+.025
3	.435	**.455**	+.020
4	.441	**.465**	+.024

are descriptive because the subsets differ between policies, so we do not treat them as the causal budget comparison. Read together with the enforced-cap results, however, they show both sides of the behavior: Pareto-GRPO is better when the budget is fixed and learns to reach high accuracy with fewer calls when it chooses the budget itself.

Table 7: **Qwen2.5-7B EM conditioned on realized search usage.** The enforced-cap experiment in Table 6 is the controlled budget comparison.

Method	$n = 0$	$n \leq 1$	$n \leq 2$	$n \leq 3$	$n \leq 4$
Outcome-only	.121	.243	.308	.338	.350
Pareto-GRPO	**.236**	**.379**	**.433**	**.457**	**.465**

The construction is also not tied to search count. Table 8 replaces the second objective by generated tokens, measured latency/API cost, or a hybrid. All variants retain most of the accuracy gain while reducing the cost they directly optimize. Latency is measured on fixed service and hardware, so its percentage is an in-environment measurement rather than a universal systems claim.

Table 8: **Alternative cost objectives on Qwen2.5-7B.** Cost changes are relative to outcome-only GRPO in the named unit.

Second objective	EM	Cost change	HV
Search count	**.465**	-33.6% searches	**.286**
Generated tokens	.460	-17% tokens	.280
Latency/API cost	.455	-12% latency	.274
Hybrid search + tokens	.460	$-32\%/-14\%$	.283

Nor is Pareto ranking restricted to two rewards. We add token efficiency to correctness and search efficiency and fix normalization and the 3D-HV reference point across methods. Table 9 shows that Pareto-GRPO improves every reported coordinate over tuned three-reward weighted and GDPO-style controls. We do not extrapolate this result to arbitrarily many objectives; it establishes that the mechanism survives the first non-trivial increase in dimension.

Strictly monotone maps of the same search count provide a sharper test of the ordinal claim. The transforms $\exp(-n/B)$, $1/(1+n)$, and $1-n/B$ yield 100% identical dominance graphs and ranks and the same $.465/1.780/.286$ result. In contrast, Table 8 changes what is being optimized and therefore need not agree. Appendix F gives the full audit.

Table 9: **Three-objective extension.** Relative 3D HV is normalized to the GDPO-style control.

Method	EM	Search	Token reduction	Rel. 3D HV
Weighted sum	.425	1.95	12%	.97
GDPO-style	.435	1.90	15%	1.00
Three-objective Pareto-GRPO	**.460**	**1.82**	**18%**	**1.10**

4.5 Observed training behavior

The rank statistics in Figure 4 evolve with the policy: average and maximum rank fall as the sampled groups concentrate near the empirical frontier, while the efficiency trace improves. This is the pattern expected if the shaped update removes dominated over-searching behavior rather than merely shrinking every trajectory.

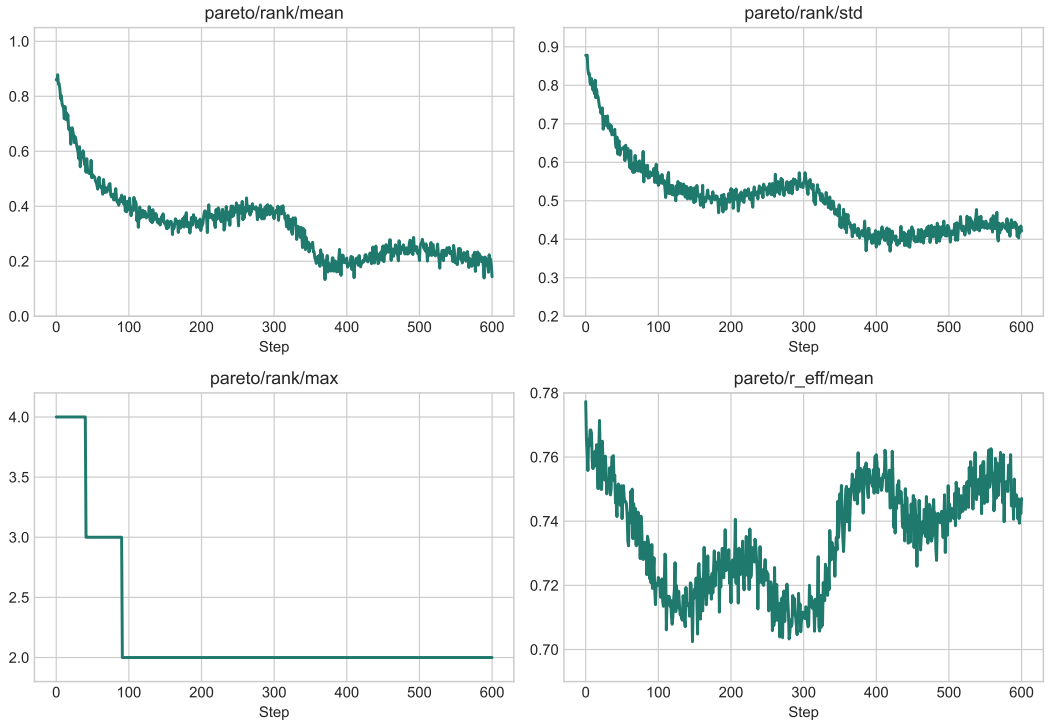

Figure 4: **Pareto diagnostics over training.** Mean, standard deviation, and maximum Pareto rank together with mean efficiency track the ordering process directly.

Figure 5 complements the aggregate diagnostics with observed rollouts. On easy questions, Pareto-GRPO terminates after obtaining sufficient evidence; on harder and multi-hop questions, it retains the additional searches needed for a correct answer. The behavior is selective rather than a blanket search penalty, matching the hard-cap and cost-substitution results.



Figure 5: **Observed rollout cases.** GRPO and Pareto-GRPO on easy, medium, and multi-hop examples, including final answers and logged search counts.

4.6 Scope of the evidence

The experiments establish a better reward-vector-to-advantage interface for trajectory-level search-agent RL; they do not establish a global hypervolume guarantee or token-level credit assignment. Within that scope, the evidence is consistent across datasets, scales, seeds, enforced constraints, cost definitions, and objective dimension. The appendix retains the original comparison tables, budget curves, sensitivity study, and additional diagnostics.

5 Conclusion

Pareto-GRPO turns a reward vector into a prompt-local ordinal advantage without imposing one global exchange rate between correctness and cost. It outperforms tuned scalar, prompt-local, and decoupled-normalization baselines while using fewer searches and sharply reducing seed variance. Source isolation identifies the true Pareto order as the active ingredient; enforced caps, alternative costs, and three objectives confirm the result under realistic changes. When a verifier leaves agent trajectories tied, the comparison rule matters as much as the rewards being compared.

AI use statement

Generative AI tools were used to assist with manuscript organization, prose editing, summarization of author-provided experimental results, and LaTeX formatting. They were not used to generate experimental measurements. The authors checked all AI-assisted text against the experiment records, verified the mathematical statements and citations, and take responsibility for the complete submission.

Reproducibility statement

Appendix D documents the datasets, controlled baselines, model and retrieval setup, search accounting, evaluation metrics, validation-only hyperparameter selection, and compute budget. Appendix C contains complete proofs. Appendices E–F give the sweeps and objective extensions, and Appendices J–M retain the diagnostics, ablations, sensitivity analyses, and additional scale results.

REFERENCES

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Andreas Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards, 2019. URL https://arxiv.org/abs/1806.07857.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. Research: Learning to reason with search for llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2503.19470.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International conference on machine learning*, pp. 10764–10799. PMLR, 2023.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), 2022. URL https://arxiv.org/abs/2103.09568.

Bowen Jin, Jinsung Yoon, Priyanka Kargupta, Sercan O. Arik, and Jiawei Han. An empirical study on reinforcement learning for reasoning-search interleaved LLM agents, 2025a. URL https://arxiv.org/abs/2505.15117.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning, 2025b. URL https://arxiv.org/abs/2503.09516.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. URL https://doi.org/10.1162/tacl_a_00276.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models, 2025. URL https://arxiv.org/abs/2501.05366.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization, 2026. URL https://arxiv.org/abs/2601.05242.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback, 2021. URL https://arxiv.org/abs/2112.09332.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

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, 2024. URL https://arxiv.org/abs/2305.18290.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

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. URL https://arxiv.org/abs/2302.04761.

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, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of LLMs without searching, 2025. URL https://arxiv.org/abs/2505.04588.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition, 2022. URL https://arxiv.org/abs/2108.00573.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, 2023. URL https://arxiv.org/abs/2212.10509.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting LLMs search ability via step-wise proximal policy optimization, 2025. URL https://arxiv.org/abs/2505.15107.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Teng Xiao, Yige Yuan, Hamish Ivison, Huaisheng Zhu, Faeze Brahman, Nathan Lambert, Pradeep Dasigi, Noah A. Smith, and Hannaneh Hajishirzi. Meta-reinforcement learning with self-reflection for agentic search, 2026. URL https://arxiv.org/abs/2603.11327.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. URL https://aclanthology.org/D18-1259/.

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*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2205.10625.

A Limitations and Discussion

Our evidence concerns trajectory-level credit assignment for search agents. Alternative costs and three-objective experiments broaden the result beyond one two-dimensional reward, but they do not establish behavior for large objective sets, token-level credit assignment, coding agents, or multimodal interaction. Pareto rank also becomes less discriminative as objective dimension grows. The first-order result characterizes the expected on-policy estimator for a fixed behavior policy; it is not a finite-step guarantee for clipped optimization. Finally, faithful reproduction depends on aligning cost accounting with the environment rather than inferring tool use from generated text after training.

B Broader Impact

The main positive impact is improved efficiency for search-augmented language agents: better search control can reduce unnecessary tool use, latency, and serving cost. The main potential risk is that stronger search agents may also make information-seeking misuse more effective. We therefore view this method as a capability-improving technique that should be paired with ordinary deployment safeguards such as access control, logging, and human oversight in high-stakes settings.

C Formal Statements and Proofs

Setup. Fix a prompt and let the sampled trajectory group be

$$\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_G\}.$$

For each trajectory τ_i, let its reward vector be

$$\mathbf{r}_i = [r_{\text{acc},i}, r_{\text{eff},i}] \in \mathbb{R}^2.$$

Let the scalar reward used by the GRPO backbone be $r_i \in \mathbb{R}$. In the main experiments, $r_i = r_{\text{acc},i}$.

Definition 1 (Pareto dominance)**.** *For two trajectories $\tau_a, \tau_b \in \mathcal{T}$, we say that τ_a* dominates *τ_b, written $\tau_a \succ \tau_b$, if*

$$r_{acc,a} \geq r_{acc,b}, \qquad r_{eff,a} \geq r_{eff,b},$$

and at least one of the two inequalities is strict.

Definition 2 (Non-dominated fronts and Pareto rank)**.** *Define the first front*

$$F_0 = \{\tau_i \in \mathcal{T} : \nexists\, \tau_j \in \mathcal{T} \text{ such that } \tau_j \succ \tau_i\}.$$

Recursively, for $m \geq 1$, let

$$\mathcal{T}^{(m)} = \mathcal{T} \setminus \bigcup_{\ell=0}^{m-1} F_\ell$$

and define

$$F_m = \{\tau_i \in \mathcal{T}^{(m)} : \nexists\, \tau_j \in \mathcal{T}^{(m)} \text{ such that } \tau_j \succ \tau_i\}.$$

If $\tau_i \in F_m$, its Pareto rank is $k_i = m$.

Definition 3 (Pareto utility and advantages)**.** *For each i, define the Pareto utility*

$$u_i = \frac{G - k_i}{G}.$$

Define the scalar GRPO term

$$z_i = \frac{r_i - \mu_g}{\sigma_g + \epsilon},$$

where

$$\mu_g = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad \sigma_g = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu_g)^2}.$$

The uncentered Pareto-shaped advantage is

$$A_i = z_i + \alpha u_i.$$

The centered variant is

$$A_i^{\mathrm{c}} = z_i + \alpha(u_i - \bar{u}), \qquad \bar{u} = \frac{1}{G}\sum_{j=1}^{G} u_j.$$

Proposition 1 (First-order objective characterization)**.** *Fix a behavior policy $\bar{\pi} = \pi_{\bar{\theta}}$. For a focal trajectory and $G-1$ independently sampled companions, define*

$$\bar{A}_G^{\bar{\pi}}(\tau, x) = \mathbb{E}_{\tau_{2:G}\sim\bar{\pi}(\cdot|x)}[z(\tau, \tau_{2:G}) + \alpha u(\tau, \tau_{2:G})]$$

and $J_G(\theta;\bar{\pi}) = \mathbb{E}_x\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}[\bar{A}_G^{\bar{\pi}}(\tau, x)]$, where $\bar{A}_G^{\bar{\pi}}$ is held fixed when differentiating. Then

$$\nabla_\theta J_G(\theta;\bar{\pi})|_{\theta=\bar{\theta}} = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\nabla_{\bar{\theta}}\log\pi_{\bar{\theta}}(\tau_i|x)A_i\right].$$

Proof. The score-function identity gives

$$\nabla_\theta J_G(\theta;\bar{\pi})|_{\bar{\theta}} = \mathbb{E}_{x,\tau_1\sim\bar{\pi}}[\nabla_{\bar{\theta}}\log\pi_{\bar{\theta}}(\tau_1|x)\bar{A}_G^{\bar{\pi}}(\tau_1, x)].$$

Expanding the conditional expectation in $\bar{A}_G^{\bar{\pi}}$ yields the expected contribution of the first member of an i.i.d. group. Exchangeability makes every member's expected contribution identical, so it equals the expectation of their group average. This is the stated estimator. The statement concerns the unclipped on-policy direction at $\bar{\theta}$; clipping, KL regularization, and a finite parameter update are outside the equality. □

Proposition 2 (Coordinate-wise monotone invariance)**.** *Let each reward coordinate $r^{(d)}$ be replaced by $h_d(r^{(d)})$, where h_d is strictly increasing. The dominance graph, all non-dominated fronts, ranks k_i, and utilities u_i are unchanged.*

Proof. For every coordinate and pair a, b, strict monotonicity preserves both $r_a^{(d)} \geq r_b^{(d)}$ and $r_a^{(d)} > r_b^{(d)}$. It therefore preserves every dominance edge. Non-dominated sorting is a deterministic function of that graph, which proves the remaining claims. □

Proposition 3 (Dominance implies strictly lower rank)**.** *If $\tau_a \succ \tau_b$, then $k_a < k_b$.*

Proof. Assume $\tau_a \succ \tau_b$.

We first show that τ_a and τ_b cannot belong to the same front. Suppose for contradiction that both belong to F_m for some $m \geq 0$. By definition of F_m, every element of F_m is non-dominated within $\mathcal{T}^{(m)}$. Since $\tau_a \in F_m \subseteq \mathcal{T}^{(m)}$ and $\tau_b \in F_m \subseteq \mathcal{T}^{(m)}$, the relation $\tau_a \succ \tau_b$ implies that τ_b is dominated by an element of $\mathcal{T}^{(m)}$, namely τ_a. This contradicts the definition of F_m.

So $k_a \neq k_b$. It remains to show that k_a cannot be larger than k_b. Suppose for contradiction that $k_a > k_b$. Then $\tau_b \in F_{k_b} \subseteq \mathcal{T}^{(k_b)}$. Since fronts are removed in increasing order, $k_a > k_b$ implies $\tau_a \in \mathcal{T}^{(k_b)}$ as well. But $\tau_a \succ \tau_b$, so within the set $\mathcal{T}^{(k_b)}$ the trajectory τ_b is dominated by τ_a. Therefore τ_b cannot belong to F_{k_b}, contradicting the definition of F_{k_b}.

Both contradictions establish that $k_a < k_b$. □

Proposition 4 (Correctness-order preservation). *Suppose a group contains both reward values, let $p \in (0,1)$ be its fraction of correct trajectories, and let τ_a be correct while τ_b is incorrect. For either centered or uncentered shaping, $A_a > A_b$ is guaranteed whenever*

$$\alpha < \frac{1}{(\sqrt{p(1-p)} + \epsilon)(1 - 1/G)}.$$

Proof. For binary correctness, $\mu_g = p$ and $\sigma_g = \sqrt{p(1-p)}$, hence

$$z_a - z_b = \frac{1}{\sqrt{p(1-p)} + \epsilon}.$$

Utilities lie in $[1/G, 1]$, so $u_a - u_b \geq -(1 - 1/G)$. Centering cancels in a pairwise difference. Therefore

$$A_a - A_b \geq \frac{1}{\sqrt{p(1-p)} + \epsilon} - \alpha(1 - 1/G),$$

which is positive under the stated condition. For $G = 5$ and negligible ϵ, the right-hand threshold is at least 2.5 because $\sqrt{p(1-p)} \leq 1/2$; the experimental value $\alpha = 0.1$ is well inside this sufficient range. □

Corollary 1 (Dominance implies larger utility). *If $\tau_a \succ \tau_b$, then $u_a > u_b$.*

Proof. By Proposition 3, $k_a < k_b$. Since

$$u_i = \frac{G - k_i}{G}$$

is a strictly decreasing function of k_i, it follows that $u_a > u_b$. □

Proposition 5 (Tie-breaking under equal scalar reward). *Suppose $r_a = r_b$ for two trajectories in the same prompt group.*

1. *If $k_a < k_b$, then $A_a > A_b$.*
2. *If $k_a < k_b$, then $A_a^{\mathrm{c}} > A_b^{\mathrm{c}}$.*

Proof. Since $r_a = r_b$, both trajectories use the same group mean μ_g and group standard deviation σ_g. Therefore

$$z_a = \frac{r_a - \mu_g}{\sigma_g + \epsilon} = \frac{r_b - \mu_g}{\sigma_g + \epsilon} = z_b.$$

For the uncentered advantage,

$$A_a - A_b = (z_a + \alpha u_a) - (z_b + \alpha u_b) = \alpha(u_a - u_b).$$

Because $k_a < k_b$, Corollary 1 implies $u_a > u_b$, hence $u_a - u_b > 0$. Since $\alpha > 0$, we conclude $A_a - A_b > 0$, so $A_a > A_b$.

For the centered advantage,

$$A_a^{\mathrm{c}} - A_b^{\mathrm{c}} = \big(z_a + \alpha(u_a - \bar{u})\big) - \big(z_b + \alpha(u_b - \bar{u})\big) = \alpha(u_a - u_b).$$

The same argument yields $A_a^{\mathrm{c}} - A_b^{\mathrm{c}} > 0$, so $A_a^{\mathrm{c}} > A_b^{\mathrm{c}}$. □

Corollary 2 (All-equal scalar rewards reduce to Pareto ordering). *If $r_1 = r_2 = \cdots = r_G$, then for all i,*

$$z_i = 0, \qquad A_i = \alpha u_i, \qquad A_i^{\mathrm{c}} = \alpha(u_i - \bar{u}).$$

Hence the within-group ordering induced by either A_i or A_i^{c} is exactly the ordering induced by Pareto utility.

Proof. If all scalar rewards are equal to some constant c, then

$$\mu_g = \frac{1}{G}\sum_{j=1}^{G} c = c.$$

Also,

$$\sigma_g = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(c-c)^2} = 0.$$

Therefore, for each i,

$$z_i = \frac{c-c}{0+\epsilon} = 0.$$

Substituting into the definitions of A_i and A_i^{c} yields the stated formulas. Since $\alpha > 0$, both advantages are strictly increasing functions of u_i, up to the common centering constant $\bar{u}$ in the centered case. Thus they induce the same within-group ordering as Pareto utility. □

Proposition 6 (Bounded perturbation by Pareto shaping)**.** *For every trajectory τ_i,*

$$0 < u_i \leq 1, \qquad |A_i - z_i| \leq \alpha, \qquad |A_i^{\mathrm{c}} - z_i| \leq \alpha.$$

Proof. By construction, each rank satisfies $k_i \geq 0$. Hence

$$u_i = \frac{G-k_i}{G} \leq \frac{G}{G} = 1.$$

Also, every trajectory belongs to some front generated from a group of size G, so $k_i \leq G-1$. Therefore

$$u_i = \frac{G-k_i}{G} \geq \frac{G-(G-1)}{G} = \frac{1}{G} > 0.$$

This proves $0 < u_i \leq 1$.

For the uncentered advantage,

$$A_i - z_i = \alpha u_i.$$

Taking absolute values and using $0 < u_i \leq 1$ gives

$$|A_i - z_i| = \alpha u_i \leq \alpha.$$

For the centered advantage, note first that $\bar{u}$ is the average of numbers in $(0, 1]$, so

$$0 < \bar{u} \leq 1.$$

Hence for any i,

$$-1 < u_i - \bar{u} < 1,$$

which implies

$$|u_i - \bar{u}| \leq 1.$$

Since

$$A_i^{\mathrm{c}} - z_i = \alpha(u_i - \bar{u}),$$

we obtain

$$|A_i^{\mathrm{c}} - z_i| = \alpha|u_i - \bar{u}| \leq \alpha.$$

This completes the proof. □

Proposition 7 (Zero-mean property of the centered bonus)**.** *The centered Pareto shaping term has zero empirical mean over the group:*

$$\frac{1}{G}\sum_{i=1}^{G}(u_i - \bar{u}) = 0.$$

Consequently,

$$\frac{1}{G}\sum_{i=1}^{G} A_i^{\mathrm{c}} = \frac{1}{G}\sum_{i=1}^{G} z_i.$$

Proof. By definition of $\bar{u}$,

$$\bar{u} = \frac{1}{G}\sum_{j=1}^{G} u_j.$$

Therefore,

$$\frac{1}{G}\sum_{i=1}^{G}(u_i - \bar{u}) = \frac{1}{G}\sum_{i=1}^{G} u_i - \frac{1}{G}\sum_{i=1}^{G}\bar{u}.$$

The first term equals $\bar{u}$. The second term equals

$$\frac{1}{G}\cdot G\bar{u} = \bar{u}.$$

Hence the difference is 0.

Now use

$$A_i^{\mathrm{c}} = z_i + \alpha(u_i - \bar{u}).$$

Averaging over i gives

$$\frac{1}{G}\sum_{i=1}^{G} A_i^{\mathrm{c}} = \frac{1}{G}\sum_{i=1}^{G} z_i + \alpha\frac{1}{G}\sum_{i=1}^{G}(u_i - \bar{u}) = \frac{1}{G}\sum_{i=1}^{G} z_i,$$

where the second term vanishes by the first part. □

Proposition 8 (Variance bound for centered and uncentered shaping)**.** *Let*

$$\mathrm{Var}_g(x) = \frac{1}{G}\sum_{i=1}^{G}(x_i - \bar{x})^2$$

denote empirical variance over the group. Then

$$\mathrm{Var}_g(A) = \mathrm{Var}_g(A^{\mathrm{c}})$$

and both satisfy

$$\mathrm{Var}_g(A) \le 2\,\mathrm{Var}_g(z) + \frac{\alpha^2}{2}.$$

Proof. Since

$$A_i = z_i + \alpha u_i \qquad \text{and} \qquad A_i^{\mathrm{c}} = z_i + \alpha(u_i - \bar{u}),$$

we have

$$A_i = A_i^{\mathrm{c}} + \alpha\bar{u}$$

for all i. Thus the vectors $(A_i)_{i=1}^G$ and $(A_i^{\mathrm{c}})_{i=1}^G$ differ by the same constant shift. Empirical variance is invariant under adding a constant to every coordinate, so

$$\mathrm{Var}_g(A) = \mathrm{Var}_g(A^{\mathrm{c}}).$$

It remains to bound $\mathrm{Var}_g(A^{\mathrm{c}})$. Write

$$\tilde{u}_i = u_i - \bar{u}.$$

Then

$$A_i^{\mathrm{c}} = z_i + \alpha\tilde{u}_i.$$

Therefore,

$$\mathrm{Var}_g(A^{\mathrm{c}}) = \mathrm{Var}_g(z + \alpha\tilde{u}) = \mathrm{Var}_g(z) + \alpha^2\mathrm{Var}_g(\tilde{u}) + 2\alpha\,\mathrm{Cov}_g(z, \tilde{u}),$$

where Cov_g is the empirical covariance over the group.

Because empirical variance is unchanged by centering,

$$\mathrm{Var}_g(\tilde{u}) = \mathrm{Var}_g(u).$$

Also, every u_i lies in the interval $(0, 1]$ by Proposition 6. Hence the range of u is at most 1, and Popoviciu's inequality implies

$$\mathrm{Var}_g(u) \le \frac{1^2}{4} = \frac{1}{4}.$$

Therefore

$$\mathrm{Var}_g(\tilde{u}) \le \frac{1}{4}.$$

Next, by Cauchy–Schwarz,

$$|\mathrm{Cov}_g(z, \tilde{u})| \le \sqrt{\mathrm{Var}_g(z)\mathrm{Var}_g(\tilde{u})}.$$

Thus

$$2\alpha\,\mathrm{Cov}_g(z, \tilde{u}) \le 2\alpha\sqrt{\mathrm{Var}_g(z)\mathrm{Var}_g(\tilde{u})}.$$

Apply the elementary inequality $2ab \le a^2 + b^2$ with

$$a = \sqrt{\mathrm{Var}_g(z)}, \qquad b = \alpha\sqrt{\mathrm{Var}_g(\tilde{u})}.$$

We obtain

$$2\alpha\sqrt{\mathrm{Var}_g(z)\mathrm{Var}_g(\tilde{u})} \le \mathrm{Var}_g(z) + \alpha^2\mathrm{Var}_g(\tilde{u}).$$

Substituting into the variance expansion gives

$$\mathrm{Var}_g(A^{\mathrm{c}}) \le 2\,\mathrm{Var}_g(z) + 2\alpha^2\mathrm{Var}_g(\tilde{u}).$$

Using $\mathrm{Var}_g(\tilde{u}) \le 1/4$, we conclude

$$\mathrm{Var}_g(A^{\mathrm{c}}) \le 2\,\mathrm{Var}_g(z) + 2\alpha^2 \cdot \frac{1}{4} = 2\,\mathrm{Var}_g(z) + \frac{\alpha^2}{2}.$$

Finally, since $\mathrm{Var}_g(A) = \mathrm{Var}_g(A^{\mathrm{c}})$, the same bound holds for $\mathrm{Var}_g(A)$. □

D Experimental Setup Details

D.1 Tasks and data

We focus on search-augmented question answering with both single-hop and multi-hop reasoning. Our main benchmarks are:

- **Single-hop QA**: Natural Questions (NQ) (Kwiatkowski et al., 2019).
- **Multi-hop QA**: HotpotQA (Yang et al., 2018).

We train search-interleaved language agents using the Search-R1 framework (Jin et al., 2025b) and evaluate on a seven-benchmark suite (NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultihopQA, MuSiQue, and Bamboogle) to cover both single-hop and multi-hop settings (Joshi et al., 2017; Kwiatkowski et al., 2019; Yang et al., 2018; Trivedi et al., 2022).

D.2 Baselines

To isolate the value of Pareto ranking rather than generic reward shaping, we compare against:

- **GRPO**: standard outcome-only group optimization (Shao et al., 2024).
- **GRPO + search penalty**: scalar reward with an explicit efficiency penalty, following the common scalarization view in multi-objective RL (Hayes et al., 2022).
- **Weighted-sum multi-objective**: fixed scalarization of correctness and efficiency (Roijers et al., 2013; Hayes et al., 2022).
- **Prompt-local scalarization**: a scalar correctness–efficiency weight selected and applied within the same prompt-group training protocol.

- **GDPO-style**: reward-wise decoupled normalization for multiple rewards (Liu et al., 2026).
- **Random scalarization**: per-batch or per-group sampled objective weights as a stochastic preference baseline motivated by multi-objective optimization practice (Hayes et al., 2022).

For the final controlled comparison, we sweep scalar penalty over $\{0, .02, .05, .10, .20, .30\}$, weighted sum over $\{0, .02, .05, .10, .20, .30, .50\}$, prompt-local scalarization over $\{.02, .05, .10, .20, .30, .50\}$, GDPO-style weighting over $\{.05, .10, .20, .50, 1.0\}$, and Pareto α over $\{.05, .10, .20\}$. Validation hypervolume selects .10, .20, .10, .20, and .10, respectively. No test or held-out benchmark result enters model selection. We additionally include a no-search direct-answer baseline for the realized-budget analysis and random scalarization as a stochastic control.

Why the Search-R1 numbers differ across tables. The paper uses two comparison protocols. The main result, stability, budget, and ablation tables use an in-house outcome-only GRPO control under the same environment, model family, and evaluation pipeline as Pareto-GRPO. Table 15, by contrast, is a cross-paper comparison against numbers reported in prior work under the same benchmark suite and training-data setting. The strongest Search-R1 numbers in the original Search-R1 line are associated with a PPO-style training route rather than the GRPO control used for our internal ablations, so the Search-R1 row in the aligned comparison does not exactly match the internal baseline used elsewhere in the paper.

Precise definitions of the scalarized baselines. Let $r_{\text{acc}}(\tau_i) \in \{0, 1\}$ be correctness, let $n_{\text{search}}(\tau_i)$ be the number of valid search calls, and let $r_{\text{eff}}(\tau_i) = \exp(-n_{\text{search}}(\tau_i)/B)$ be the efficiency score used elsewhere in the paper. The *scalar search penalty* baseline uses a single scalar reward of the form

$$r_i^{\text{pen}} = r_{\text{acc}}(\tau_i) - \lambda \frac{n_{\text{search}}(\tau_i)}{B},$$

with λ selected by the sweep above. The *weighted-sum* baseline uses a fixed scalarization over correctness and efficiency,

$$r_i^{\text{ws}} = (1 - \lambda)\, r_{\text{acc}}(\tau_i) + \lambda\, r_{\text{eff}}(\tau_i),$$

with λ selected by the same protocol. The *random scalarization* baseline uses the same scalarization family but resamples the scalarization weight during training, i.e.

$$r_i^{\text{rand}} = (1 - \lambda_i)\, r_{\text{acc}}(\tau_i) + \lambda_i\, r_{\text{eff}}(\tau_i),$$

where λ_i is randomly varied across training groups or batches. We include this row as a stochastic scalarization control rather than as a separately optimized preference-learning method, so the exact sampling schedule should be viewed as a control-setting detail rather than as a scientific contribution of the paper.

D.3 Evaluation metrics

We report performance, stability, efficiency, frontier-aware, and diagnostic metrics. Performance is measured by exact match (EM). Stability is measured by the mean and standard deviation across seeds, collapse rate, and worst-seed performance. Efficiency is measured by the average number of counted search actions, the average number of counted search actions on correct examples, and search calls at matched accuracy. Frontier-aware metrics include conditional accuracy under realized search budgets and hypervolume over the empirical accuracy-cost frontier, computed in the normalized space $(\text{EM}, 1 - \min(\overline{n}_{\text{search}}, B)/B)$ with reference point $(0, 0)$. Diagnostic metrics include the mean, standard deviation, and maximum Pareto rank, frontier size, dominated count, and optionally advantage variance.

D.4 Implementation details

We use the same search environment, rollout budget, model family, and training schedule across methods unless otherwise stated. Our implementation builds on Search-R1 (Jin et al., 2025b) and replaces the GRPO outcome advantage with Pareto-ranked group optimization. We sample $G = 5$ trajectories per prompt, set the maximum search budget to $B = 4$, use two active search-interaction steps followed by a terminal generation pass, and use top-$k = 3$ retrieval unless noted otherwise.

Table 10: **Selected and cost-aggressive sweep endpoints.** Triplets are (EM, average search, HV) on the common seven-dataset evaluation after validation-only selection.

Family	Grid	Selected setting	High endpoint
Scalar penalty	0–.30	.10: .405/1.980/.223	.30: .374/1.490/.211
Weighted sum	0–.50	.20: .423/1.910/.241	.50: .389/1.310/.214
Prompt-local	.02–.50	.10: .441/1.850/.258	.50: .371/1.100/.204
GDPO-style	.05–1.0	.20: .438/1.880/.255	1.0: .401/1.400/.226
Pareto-GRPO	.05–.20	.10: .465/1.780/.286	.20: .458/1.710/.279

Table 11: **Alternative cost objectives on Qwen2.5-7B.** Cost change is relative to outcome-only GRPO in the named unit.

Second objective	EM	Cost change	HV
Search count	.465	-33.6% searches	.286
Generated tokens	.460	-17% tokens	.280
Latency/API cost	.455	-12% latency	.274
Hybrid search + tokens	.460	$-32\%/-14\%$	.283

The efficiency reward is $r_{\text{eff}} = \exp(-n_{\text{search}}/B)$ with counts clamped at B. A "valid search" is an implementation-level statistic: 'valid_search_stats' increments when the parser emits a well-formed search action during rollout, including the terminal pass used to finish generation. This exact quantity feeds reward computation, tables, and figures; it is not reconstructed from text after generation. We apply KL regularization to a fixed reference policy with coefficient 0.001. The Pareto rank weight is $\alpha = 0.1$, and the main experiments use Eq. 3; the centered form appears in Appendix L. Runs use $8\times$ NVIDIA A800 80GB GPUs. Baseline sweeps require approximately 1,000–1,200 GPU-hours per family, compared with about 800 GPU-hours for Pareto-GRPO including final runs, under the same early-stopping rule. Five seeds are used for the controlled comparison. Setting $\alpha = 0$ exactly recovers the outcome-only GRPO path.

E Controlled Baseline Sweeps

All families use validation hypervolume for selection and the same early-stopping rule. Table 10 reports both the selected setting and the most cost-aggressive endpoint from each sweep. The drop at the high-weight endpoint shows why a narrow or untuned comparison can misrepresent scalarization: every family can reduce search further by sacrificing a large amount of exact match. The final controlled rows compare the validation-selected trade-offs, not arbitrarily matched weights.

F Alternative Costs and Three Objectives

Table 11 changes the second objective while leaving the training protocol fixed. Token reduction is measured against outcome-only GRPO; latency is measured on the same service and hardware. The latency result should not be transferred to a different serving stack, but it verifies that the ordering accepts a directly measured operational cost.

For the three-objective experiment, rewards are correctness, search efficiency, and token efficiency. All methods use the same fixed normalization and the same reference point for relative 3D hypervolume.

Monotone efficiency transforms. On the same rollout counts, $\exp(-n/B)$, $1/(1+n)$, and $1-n/B$ induce identical pairwise orderings. Their dominance graphs and ranks agree on 100% of sampled groups and, as Proposition 2 predicts, all three runs produce the same $.465$ EM, 1.780 searches, and $.286$ HV when no other component changes. Non-monotone or semantically different costs need not agree; the alternative-cost table tests that harder case.

Table 12: **Three-objective extension.** Relative 3D HV is normalized so that the GDPO-style baseline equals 1.00.

Method	EM	Search	Token reduction	Relative 3D HV
Three-reward weighted sum	.425	1.95	12%	.97
Three-reward GDPO-style	.435	1.90	15%	1.00
Three-objective Pareto-GRPO	**.460**	**1.82**	**18%**	**1.10**

Table 13: **Stability and efficiency across model scales.** Correct Search is the mean search count on correctly answered examples; Collapse reports unstable runs.

Backbone	Method	Avg. Search	Correct Search	Seed Std.	Worst Seed	Collapse
Qwen2.5-7B	Outcome-only GRPO	2.680	2.440	.061	.271	2/5
Qwen2.5-7B	Pareto-GRPO	**1.780**	**1.460**	**.010**	**.454**	**0/5**
Qwen2.5-3B	Outcome-only GRPO	2.570	2.310	.052	.247	2/5
Qwen2.5-3B	Pareto-GRPO	**1.960**	**1.700**	**.024**	**.329**	**0/5**

G Original Efficiency and Budget Analyses

The following analyses from the original submission are retained in full. They complement the stricter controlled comparisons in the main text.

H Contextual Cross-Paper Comparison

Table 15 retains the original contextual comparison. Its rows come from prior papers rather than our controlled reruns, so the claims in the main text rely on Table 2 instead.

I Extended Related Work

Tool-using and search-based agents. Recent language agents interleave reasoning with external actions such as search, browsing, and program execution (Yao et al., 2023; Schick et al., 2023; Nakano et al., 2021; Gao et al., 2023; Shinn et al., 2023; Asai et al., 2023). Search-specific RL work further shows that models can learn to issue and refine search queries through policy optimization, but these systems are sensitive to reward design, model initialization, and retrieval noise (Jin et al., 2025b;a; Sun et al., 2025; Trivedi et al., 2023). Our work builds on this line and focuses specifically on stabilizing search-augmented RL.

Group-based RL for language models. Group-relative optimization has emerged as a scalable alternative to critic-heavy RL for language models (Schulman et al., 2017; Ouyang et al., 2022; Rafailov et al., 2024; Shao et al., 2024). It is particularly attractive in verifiable domains, but standard outcome-only formulations provide only coarse supervision when multiple trajectories share the same final correctness. Our method keeps the GRPO-style training setup and changes only how group preferences are constructed.

Credit assignment under long horizons. Our work is also related to temporal and structured credit assignment. Delayed rewards are known to create high-variance learning signals in RL (Arjona-Medina et al., 2019), and recent reasoning work highlights that multiple trajectories can solve the same problem while differing substantially in their intermediate structure (Wei et al., 2022; Wang et al., 2023; Zhou et al., 2023). We view agentic search RL as an especially difficult version of this problem because trajectories may share the same final answer while differing greatly in tool-use cost and evidence quality.

Multi-objective RL and Pareto optimization. Multi-objective RL provides a natural framework for handling conflicting criteria (Roijers et al., 2013; Hayes et al., 2022; Deb et al., 2002). Our setting differs from preference-conditioned or policy-set approaches in MORL: we do not aim to learn an

Table 14: **Conditional accuracy under realized search usage.** Each entry conditions on trajectories whose logged count is at most k; unlike Table 6, this does not enforce a decoding cap.

Backbone	Method	$n = 0$	$n \leq 1$	$n \leq 2$	$n \leq 3$	$n \leq 4$
Qwen2.5-7B	Outcome-only GRPO	.121	.243	.308	.338	.350
Qwen2.5-7B	Pareto-GRPO	**.236**	**.379**	**.433**	**.457**	**.465**
Qwen2.5-3B	Outcome-only GRPO	.109	.214	.274	.300	.312
Qwen2.5-3B	Pareto-GRPO	**.171**	**.302**	**.356**	**.386**	**.402**

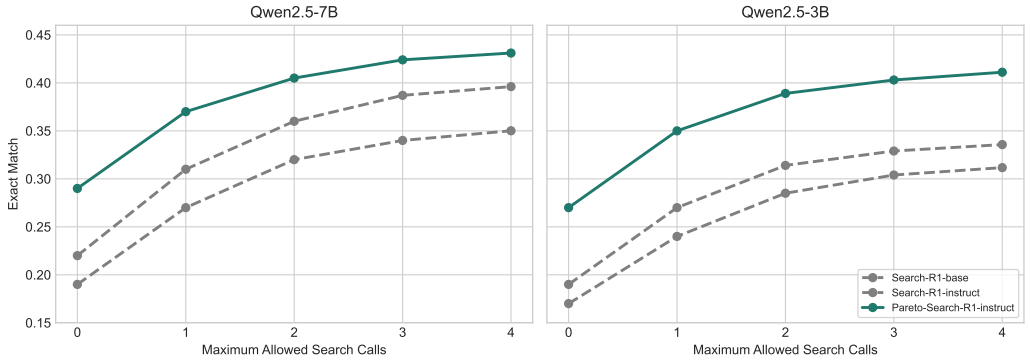

Figure 6: **Realized-budget curves.** Conditional EM versus logged search usage. These descriptive curves are complemented by the enforced-cap experiment in Table 6.

entire family of policies. Instead, we use Pareto ranking inside each prompt group as a lightweight local comparison signal for stabilizing agentic RL under correctness-efficiency trade-offs.

J Mechanistic Diagnostics

Under outcome-only GRPO, within-group ties are common in search QA because the correctness reward is binary. Multiple trajectories in the same group frequently share the same scalar label ($r = 1$ for correct or $r = 0$ for incorrect), even when their search costs differ dramatically. By introducing r_{eff} and Pareto ranking, Pareto-GRPO breaks these ties with a local ordering based on efficiency.

Over-searching trajectories are harmful for at least two reasons: they increase tool cost and expose the model to more noisy observations (Jin et al., 2025a; Sun et al., 2025). If a trajectory is both less accurate and less efficient than another trajectory for the same prompt, it is dominated and receives a lower Pareto preference. Moreover, because the Pareto bonus is bounded by group size and weighted by a small coefficient α, it refines the relative ordering of trajectories without replacing the original objective. Figure 8 tracks how group-wise rank statistics evolve during training. Figure 9 examines whether the learned policy uses search selectively rather than uniformly suppressing tool calls. Figure 10 summarizes the resulting shift in the empirical accuracy-efficiency frontier.

K Main Ablations and Sensitivity

How to interpret the $\alpha = 0$ control. In the advantage formula, setting $\alpha = 0$ removes the explicit rank bonus and exactly recovers the outcome-only GRPO baseline under the same training entry point and evaluation pipeline. The $\alpha = 0$ row is therefore a path-matched degeneration check rather than a distinct empirical result. Outcome-only GRPO versus scalarized baselines measures the effect of adding an efficiency-aware scalar objective, outcome-only GRPO versus full Pareto-GRPO measures the contribution of Pareto-ranked shaping, and the centered-versus-uncentered comparison shows that the gain is not tied to one particular normalization of the bonus.

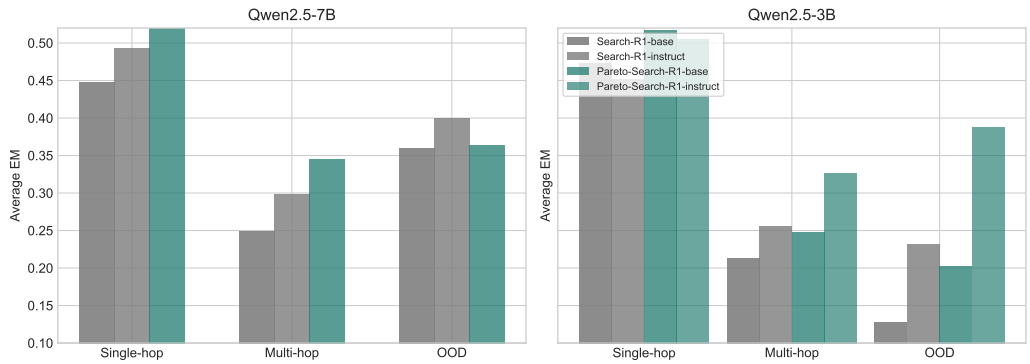

Figure 7: **In-domain and out-of-domain aggregates.** Average exact match for base and instruction-tuned Search-R1 and Pareto variants, separated into single-hop, multi-hop, and OOD groups at both model scales.

Table 15: **Published results under the same seven-benchmark suite and NQ+HotpotQA training setup.** Prior rows follow Xiao et al. (2026); they are contextual, not path-matched.

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-3B								
Direct Inference	.106	.288	.108	.149	.244	.020	.024	.134
Search-o1 (Li et al., 2025)	.238	.472	.262	.221	.218	.054	.320	.255
ReSearch (Chen et al., 2025)	.427	.597	.430	.305	.272	.074	.115	.304
Search-R1 (Jin et al., 2025b)	.462	.622	.456	.326	.310	.077	.176	.347
PPRM	.423	.565	.411	.353	.340	.127	.280	.357
StepResearch (Wang et al., 2025)	.446	.615	.456	.373	.338	.105	.325	.380
Pareto-GRPO	**.477**	**.637**	**.463**	.372	**.387**	**.143**	**.334**	**.402**
Qwen2.5-7B								
Direct Inference	.134	.408	.140	.183	.250	.031	.120	.181
Search-o1 (Li et al., 2025)	.151	.443	.131	.187	.176	.058	.296	.206
ReSearch (Chen et al., 2025)	.366	.605	.391	.378	.386	.166	.376	.381
Search-R1 (Jin et al., 2025b)	.459	.632	.449	.439	.387	.181	.400	.421
PPRM	.458	.610	.437	.386	.355	.147	.355	.393
StepResearch (Wang et al., 2025)	.473	.636	.431	.439	.418	.205	.435	.434
Pareto-GRPO	**.506**	**.673**	**.479**	**.472**	**.443**	**.225**	**.460**	**.465**

Why rank shaping does not usually dominate. Two properties limit over-correction. First, the rank term is bounded: $u_i = (G - k_i)/G \in [1/G, 1]$, so even in all-correct or all-wrong groups it remains a small additive preference rather than an unbounded reward channel. In particular, an incorrect but efficient trajectory can appear on the first Pareto front only when no correct trajectory in the same group dominates it on both objectives; in such cases the Pareto term acts as a relative preference among failures rather than a replacement for correctness supervision. Second, the main experiments use a small coefficient $\alpha = 0.1$, and Table 16 shows that performance remains stable across nearby values of α and across group sizes $G \in \{4, 5, 8\}$. That said, the direction of the rank signal still depends on the fidelity of search accounting. If the environment systematically under-counts failed or malformed search actions, the rank ordering can become biased toward trajectories that appear cheap only because search usage was not recorded correctly. We therefore report both budget-conditioned accuracy and search-at-matched-accuracy metrics rather than relying on one efficiency statistic alone.

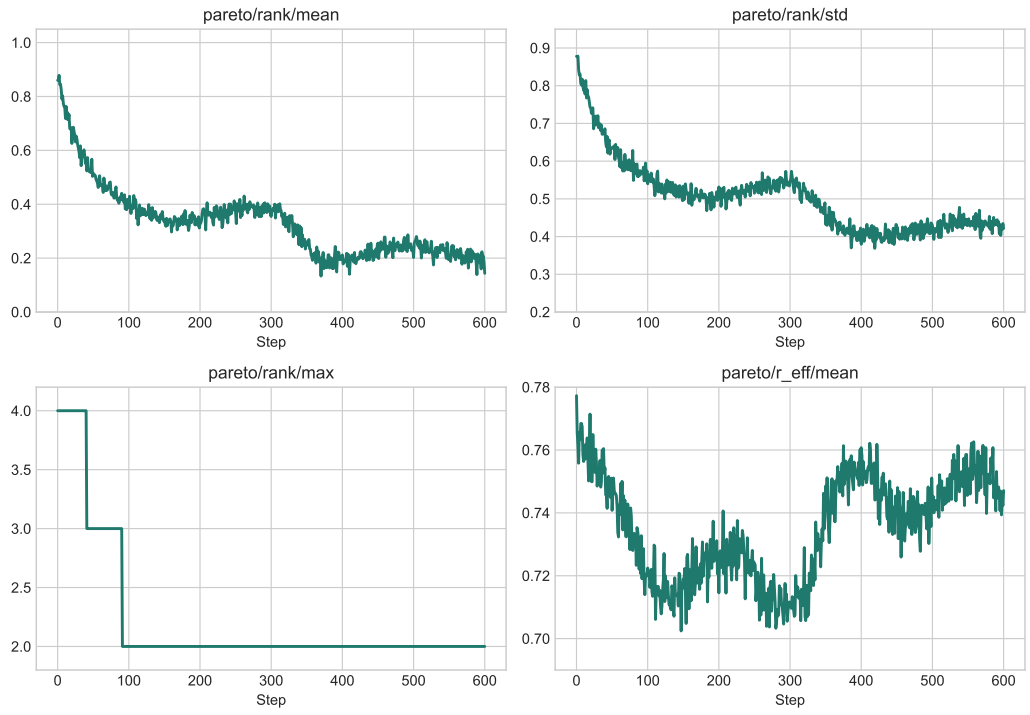

Figure 8: **Pareto diagnostics.** Rank statistics and efficiency over training steps characterize how sampled trajectories concentrate near the correctness-efficiency frontier.

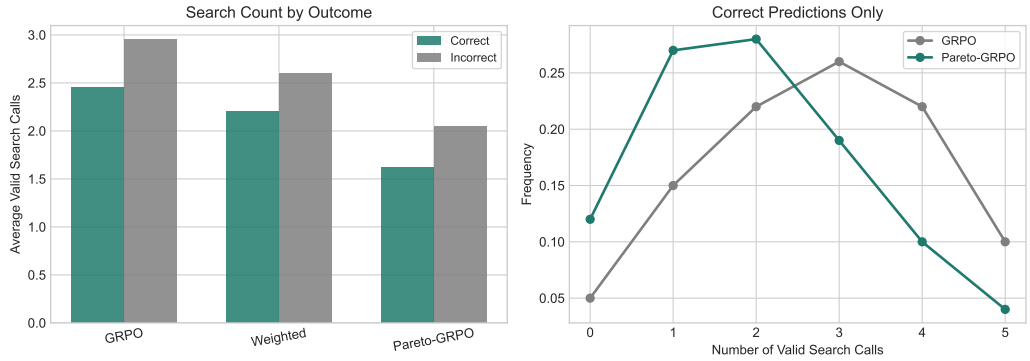

Figure 9: **Adaptive search behavior.** Search-call statistics conditioned on correctness and question type illustrate whether the policy learns selective, calibrated tool use.

L Ablation-Theory Link

Centered bonus variant. An alternative is

$$A_i^{\text{center}} = z_i + \alpha \left(u_i - \frac{1}{G} \sum_{j=1}^{G} u_j \right).$$

This preserves zero-mean shaping within a group. In the main paper we treat centered vs. uncentered bonus as an empirical ablation because stability can depend on the interaction with clipping and KL regularization.

M Additional Tables and Figures

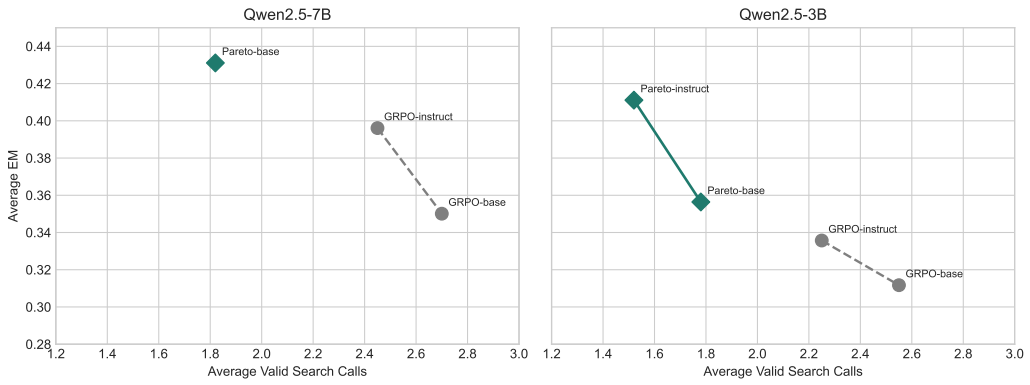

Figure 10: **Accuracy-efficiency frontier.** Each point summarizes a policy by its exact match and average search cost, visualizing the empirical correctness-efficiency frontier.

GRPO

Q: Who wrote The Old Man and the Sea?

Answer: Ernest Hemingway

Valid searches: 3

Trajectory: search -> search -> search -> answer

Pareto-GRPO

Q: Who wrote The Old Man and the Sea?

Answer: Ernest Hemingway

Valid searches: 0

Trajectory: direct answer

GRPO

Q: Which city is home to the headquarters of the company that owns Instagram?

Answer: Menlo Park

Valid searches: 4

Trajectory: search -> search -> verify -> search -> answer

Pareto-GRPO

Q: Which city is home to the headquarters of the company that owns Instagram?

Answer: Menlo Park

Valid searches: 2

Trajectory: search -> verify -> answer

GRPO

Q: Which university did the author of The Name of the Wind attend?

Answer: Unknown / wrong

Valid searches: 4

Trajectory: search -> search -> search -> search -> answer

Pareto-GRPO

Q: Which university did the author of The Name of the Wind attend?

Answer: University of Wisconsin-Stevens Point

Valid searches: 3

Trajectory: search -> refine -> verify -> answer

Figure 11: **Representative case studies.** Examples contrasting GRPO and Pareto-GRPO on easy, medium, and multi-hop questions, including final answers and valid search counts.

Table 16: **Hyperparameter sensitivity on Qwen2.5-7B.** Sensitivity to Pareto weight α, group size G, and centered vs. uncentered shaping; $\alpha = 0$ exactly recovers the outcome-only GRPO control.

Setting	EM	Avg. Search	Hypervolume	Seed Std.
$\alpha = 0$ (matches outcome-only GRPO)	0.350	2.680	0.176	0.061
$\alpha = 0.05$	0.454	1.860	0.271	0.016
$\alpha = 0.10$	**0.465**	**1.780**	**0.286**	**0.010**
$\alpha = 0.20$	0.458	1.710	0.279	0.015
group size $G = 4$	0.451	1.830	0.268	0.017
group size $G = 5$	**0.465**	**1.780**	**0.286**	**0.010**
group size $G = 8$	0.460	1.750	0.281	0.014
uncentered bonus	0.465	1.780	0.286	0.010
centered bonus	0.466	1.790	0.288	0.009

Table 17: **Original Qwen2.5-7B screening ablations, retained for completeness.** These fixed low-weight controls preceded the full sweeps in Appendix E and are not used as the strongest-baseline comparison. Avg. is the seven-benchmark mean.

Variant	Avg. EM	Avg. Search	Hypervolume	Seed Std.	Collapse Rate
Outcome-only GRPO	0.350	2.680	0.176	0.061	2 / 5
GRPO + scalar penalty, $\lambda = 0.05$	0.389	2.210	0.205	0.044	1 / 5
GRPO + weighted sum, $\lambda = 0.10$	0.414	2.020	0.226	0.032	1 / 5
GRPO + random scalarization	0.401	2.080	0.219	0.038	1 / 5
Pareto-GRPO with $\alpha = 0$ (matches outcome-only GRPO)	0.350	2.680	0.176	0.061	2 / 5
Pareto-GRPO (uncentered)	**0.465**	**1.780**	**0.286**	**0.010**	**0 / 5**
Pareto-GRPO (centered)	0.466	1.790	0.288	0.009	0 / 5

Table 18: **Original Qwen2.5-3B screening ablations, retained for completeness.** Avg. is the unweighted seven-benchmark mean; $\alpha = 0$ exactly matches outcome-only GRPO.

Variant	Avg. EM	Avg. Search	Hypervolume	Seed Std.	Collapse Rate
Outcome-only GRPO	0.312	2.570	0.143	0.052	2 / 5
GRPO + scalar penalty, $\lambda = 0.05$	0.337	2.240	0.162	0.041	1 / 5
GRPO + weighted sum, $\lambda = 0.10$	0.364	2.020	0.186	0.032	1 / 5
GRPO + random scalarization	0.352	2.100	0.175	0.039	1 / 5
Pareto-GRPO with $\alpha = 0$ (matches outcome-only GRPO)	0.312	2.570	0.143	0.052	2 / 5
Pareto-GRPO (uncentered)	**0.402**	**1.700**	**0.238**	**0.024**	**0 / 5**
Pareto-GRPO (centered)	0.403	1.710	0.240	0.023	0 / 5

Table 19: **Monotone-transform audit on Qwen2.5-7B.** When computed from the same clamped search count, the three strictly monotone efficiency maps induce identical dominance graphs, ranks, and training signals.

Efficiency definition	Rank agreement	Avg. EM	Avg. Search	HV
$\exp(-n/B)$	100%	.465	1.780	.286
$1/(1+n)$	100%	.465	1.780	.286
$1 - n/B$	100%	.465	1.780	.286