Training Optimization for Complex Reasoning Tasks in ACN: Dynamic Batch-Aware Advantage Weighting for Agentic RAG

Chen Yu[1], Li Fan[2], Wu Jie[3], Gao Weipeng[1], Ouyang Ye[1]

(1. AsiaInfo Technologies (China) Co., Ltd., Beijing 100193, China;
2. Network Optimization Center, China United Network Communications Co., Ltd., Beijing 100031, China;
3. China United Network Communications Co., Ltd., Guangzhou Branch, Guangzhou 510630, China)

DOI: 10.12142/ZTECOM.202602004

https://kns.cnki.net/kcms/detail/34.1294.TN.20260515.0959.002.html, published online May 19, 2026

Manuscript received: 2026-01-15

Abstract: With the emergence of AI-agent communication networks (ACN) in the 6G era, the efficient training of agents for complex reasoning tasks has become a critical capability for scalable ACN deployment. As a representative complex reasoning task, retrieval-augmented multi-hop question answering (e.g., agentic retrieval-augmented generation) requires agents to perform multi-step reasoning through reflection, planning, and tool-use mechanisms. However, reinforcement learning training still faces reward sparsity and sample efficiency challenges, limiting agents' rapid evolution and adaptability. We propose dynamic batch-aware advantage weighting (DB-AW), integrating two core components at the batch level: the difficulty-aware weighting component dynamically amplifies positive advantages based on long-term success rates, directing learning toward learnable yet challenging samples; and the batch filtering component removes zero-variance groups, ensuring each update contains non-zero gradient signals. Experiments show that DB-AW achieves 18%, 17%, and 15% relative improvements on Qwen2.5-7B, Qwen2.5-3B, and LLaMA3.2-3B, respectively, while improving the effective update rate from 68% to 100%, significantly reducing agent training costs. As a lightweight and reusable algorithmic module, DB-AW can be readily integrated into methods such as group relative policy optimization (GRPO), providing a practical pathway for efficient training of complex reasoning agents in ACN.

Keywords: Agentic RAG; AI-agent communication network; batch filtering; difficulty-aware weighting; multi-hop question answering; reinforcement learning

Citation (Format 1): Chen Y, Li F, Wu J, et al. Training optimization for complex reasoning tasks in ACN: dynamic batch-aware advantage weighting for agentic RAG [J]. ZTE Communications, 2026, 24(2): 26 - 32. DOI: 10.12142/ZTECOM.202602004

Citation (Format 2): Y. Chen, F. Li, J. Wu, et al., "Training optimization for complex reasoning tasks in ACN: dynamic batch-aware advantage weighting for agentic RAG," *ZTE Communications*, vol. 24, no. 2, pp. 26 - 32, Jun. 2026. doi: 10.12142/ZTECOM.202602004.

1 Introduction

As 6G networks evolve toward intelligence, AI-agent communication networks (ACN) have emerged as an infrastructure supporting massive agent interconnection[1]. Through core capabilities such as digital identity management, flexible networking, and multi-agent collaboration, ACN provides a secure and efficient communication environment for agents, empowering diverse application scenarios including intelligent customer service, network operations, and knowledge-based question answering (QA). In these applications, complex reasoning tasks (such as multi-hop question answering, multi-step planning, and code generation) pose higher requirements for agents' reasoning capabilities.

Agentic retrieval-augmented generation (Agentic RAG), as a representative complex reasoning task, provides powerful support for multi-hop question answering scenarios through agentic workflows including reflection, planning, and tool use, enabling iterative refinement and multi-step reasoning[2]. Commonly used benchmarks include HotpotQA[3], WikiMultihopQA[4], and MuSiQue[5], with NQ[6] as an open-domain reference.

Reinforcement learning (RL) is widely applied to optimize performance, but deploying and training agents for complex reasoning tasks (represented by Agentic RAG) in ACN environments faces severe challenges:

First, training efficiency bottlenecks limit agent-scale deployment. Multi-hop reasoning tasks' high difficulty leads to severe reward sparsity in outcome-based RL. Training contains numerous zero-variance batches, producing ineffective gradient updates. This not only reduces individual agent training speed but also limits the rapid iteration and evolution capabilities of agent populations in ACN, a limitation

that is particularly prominent when ACN needs to support massive parallel agent training.

Second, learning bias affects agent generalization. Policies tend to over-focus on "easy samples," failing to direct learning toward samples near the competence frontier with appropriate challenges, limiting generalization improvement. This affects agents' adaptation capabilities in the diverse task scenarios of ACN, disadvantaging flexible deployment in complex network environments.

How to efficiently integrate "difficulty-aware" and "sample efficiency improvement" optimization ideas into mainstream training algorithms like group relative policy optimization (GRPO) for complex reasoning tasks remains a worthwhile endeavor.

We propose dynamic batch-aware advantage weighting (DB-AW). DB-AW introduces two core components: difficulty-aware advantage weighting and batch filtering. Without changing reward formulation, we dynamically amplify positive advantages based on long-term historical success rates following within-group normalization, directing learning toward samples near the competence frontier. At the batch level, we identify and filter out zero-variance groups, ensuring each update contains non-zero gradient signals and significantly improving effective update steps per unit of computation.

The main contributions are summarized as follows.

1) An RL optimization method for complex reasoning tasks in ACN: Using Agentic RAG multi-hop QA as a representative scenario and addressing training challenges such as reward sparsity and learning bias, we propose DB-AW with two synergistic components:

• Difficulty-aware advantage weighting: It dynamically amplifies positive advantages based on long-term success rates after within-group normalization.

• Batch filtering: We introduce the first batch-level zero-variance group filtering mechanism for Agentic RAG scenarios. It operates without task-specific designs, maintaining lightweight and general applicability.

DB-AW optimizes GRPO training through gradient reweighting without altering reward definition.

2) Empirical validation and analysis: Under controlled setup, we validate DB-AW's effectiveness on HotpotQA, NQ, and 2Wiki. DB-AW achieves 18%, 17%, and 15% relative improvements on Qwen2.5-7B, Qwen2.5-3B, and LLaMA3.2-3B, respectively, validating cross-model generalization.

2 Related Work

2.1 Agentic RAG

Agentic RAG introduces reflection, planning, and tool use into retrieval-augmented pipelines, forming "think→retrieve →rethink" iterative loops[2]. ReAct [7], Self-RAG [8], and Reflexion[9] implement closed-loop control. Search-o1[10] achieves autonomous knowledge supplementation through Agentic RAG mechanisms.

2.2 GRPO in Agentic RAG

GRPO[11] balances "no value network" and "sample efficiency" through group relative advantages and ratio clipping. Search-R1[12] implements multi-turn alternating reasoning-retrieval. R1-Searcher[13] proposes two-stage RL: Stage-1 learns retrieval calling and Stage-2 focuses on effective utilization. Despite their demonstrated effectiveness, challenges remain, such as reward sparsity, sample distribution imbalance, and zero-gradient updates.

2.3 ACN and Multi-Agent Training Research

The emergence of ACN raises new training requirements. Research shows that multi-agent collaboration incurs high communication overhead (up to 15 times token consumption[14]) and suffers from low training efficiency. While offline MARL[15] and parameter sharing[16] make progress in multi-agent training, efficient individual agent training remains the foundation for scalable ACN deployment. This paper focuses on training optimization of individual Agentic RAG agents in ACN, laying the groundwork for multi-agent collaborative training.

2.4 Reinforcement Learning Improvement Algorithms

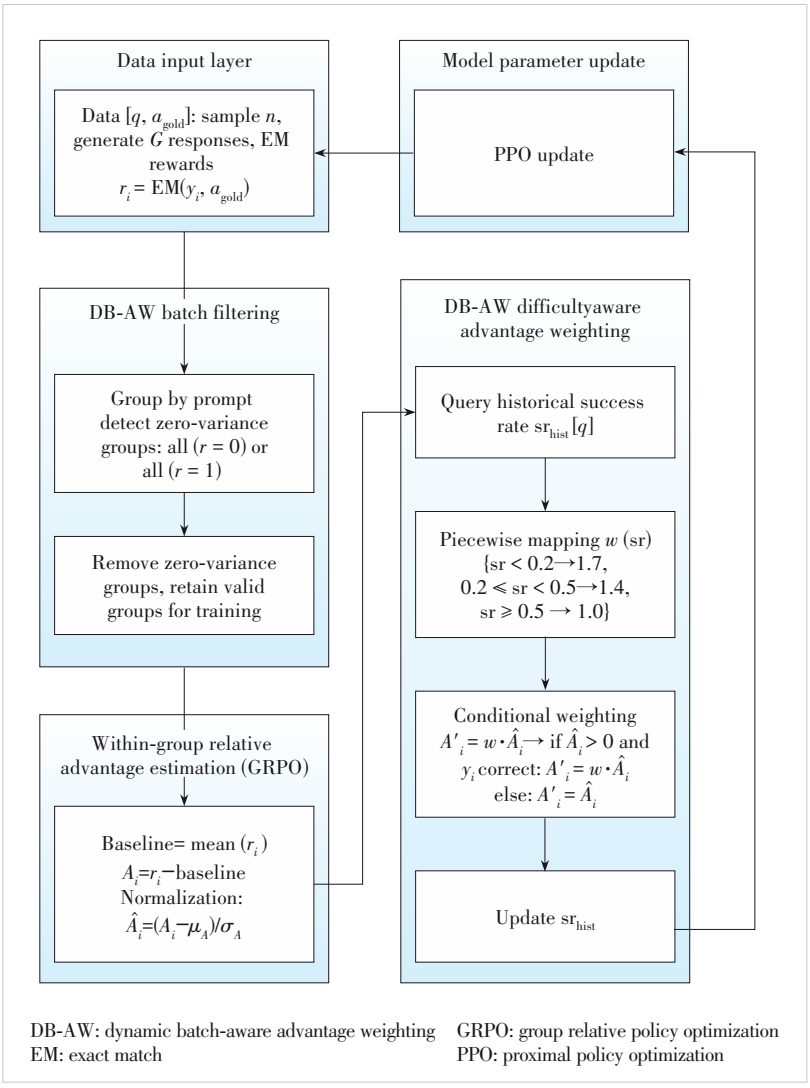
OHEM[17] and curriculum learning[18] emphasize "learnable yet challenging" samples. Decoupled clip and dynamic sampling policy optimization (DAPO)[19] integrates dynamic sampling, asymmetric clipping, and length penalties, achieving significant results in mathematical reasoning but potentially harming reasonable long-chain reasoning in Agentic RAG (e.g., HotpotQA). RLOO and REINFORCE++[20-21] improve performance from variance and robustness perspectives, while DPO and SimPO[22-23] simplify alignment through preference signals.

We propose DB-AW, introducing "difficulty-aware advantage weighting+batch filtering" at the batch level. Batch filtering draws inspiration from DAPO's zero-variance removal strategy but is tailored for Agentic RAG without imposing task-specific length penalties, therefore maintaining lightweight design and general applicability.

3 Methodology

We elaborate on DB-AW and improve training efficiency and sample utilization through batch-aware strategies within the GRPO framework. The core philosophy is to identify and filter ineffective training groups at the batch level while dynamically adjusting advantage weights based on sample difficulty, thereby directing learning toward learnable yet challenging samples and significantly improving training effectiveness without increasing computational overhead.

As shown in Fig. 1, DB-AW's complete training loop consists of five key stages: 1) data sampling and reward computa-



DB-AW: dynamic batch-aware advantage weighting
EM: exact match
GRPO: group relative policy optimization
PPO: proximal policy optimization

Figure 1. Overview of DB-AW training framework

tion; 2) batch filtering (Component 1), which removes zero-variance groups to ensure that each update contains effective gradient signals; 3) within-group advantage estimation, computing relative advantages following standard GRPO; 4) difficulty-aware weighting (Component 2), which dynamically amplifies positive advantages based on the prompt's long-term historical success rate; and 5) policy gradient update and history tracking, executing proximal policy optimization (PPO) updates using weighted advantages and statistic updates through exponential moving average (EMA).

The name DB-AW reflects three characteristics: "dynamic" refers to the dynamic adjustment of weights based on sample difficulty; "batch-aware" emphasizes batch-level optimization decisions; and "advantage weighting" indicates the method's final effect on GRPO's advantage estimation.

3.1 Problem Formulation and Base Algorithm: GRPO

1) Problem formulation: We formulate the Agentic RAG RL training task based on GRPO. Given an input sample x, the model generates interleaved structured tags: it reasons within <think> ⋯</think>; triggers retrieval through <tool call> {...} </tool call>; receives the reinjected result as <tool response>; and outputs the answer in <answer> ⋯ </answer>.

2) Loss masking: During training, we sample G responses per prompt. In all experiments, we set G = 4 due to the high interaction cost of multi-turn retrieval (each prompt requires multiple tool calls). We mask out external retrieval tokens in <tool response> and compute policy objectives only on model-generated tokens, avoiding meaningless fitting of retrieved text. In general, larger G can yield more stable group-relative advantage estimates but increases computation and tool usage.

3) Reward design: We adopt outcome-based sparse signals, using exact match (EM) as the sole outcome-based reward.

4) Relative advantages are calculated using the within-group mean as the baseline b = mean ($\{r_i\}$), and $A_i = r_i - b$. The objective function adopts GRPO's ratio clipping (PPO style, no value network) with KL regularization to monitor distribution drift.

3.2 DB-AW: Difficulty-Aware Advantage Weighting Component

1) Motivation: Dynamic weighting for learnable yet challenging samples improves sample efficiency. We adopt "weighting after within-group normalization", avoiding offset introduced by within-group de-meaning and normalization.

2) Method: Track the historical success rate sr_{hist} using the prompt key, updated via within-group average accuracy. Computation flow is shown as follows.

• Historical success rate tracking. We track sr_{hist} at the question (prompt) level using a $prompt_{key}$ (either a question ID if available, or a hash of normalized question text) as the unique identifier. The tracker is independent of retrieved evidence and random seeds.

• EMA update. After each training step, we compute $sr_{current}$ as the within-group mean accuracy for that prompt and update the tracker with an EMA: $sr_{hist}[prompt_{key}] \leftarrow \alpha\ sr_{current} + (1 - \alpha) \cdot sr_{hist}[prompt_{key}]$, where α defaults to 0.3 (recommended range: [0.2, 0.5]).

• Cold start. For a new $prompt_{key}$, we initialize sr_{hist} to 0.5, which maps to w = 1.0 (i. e., no re-weighting). Difficulty-

aware amplification is applied only after at least one update is observed for that prompt.

3) Design choice: We amplify only correct samples with positive standardized advantage. Our goal is to amplify the scarce, informative gradient signals that correspond to successful solutions on difficult prompts (e. g., sr < 0.2 for smaller backbones). These successes serve as high-quality demonstrations of correct retrieval and reasoning. We avoid additionally increasing penalties for failures on very hard prompts, which can make training unstable and overly conservative under sparse outcome rewards. Note that failed samples (reward = 0) still contribute their standard negative gradients via GRPO; DB-AW does not suppress error-correction signals. With a binary EM reward, we cannot distinguish "nearly correct" failures; finer-grained process rewards could address this and are left for future work.

a) Obtain within-group advantages $A_i = r_i - \text{mean}(\{r\})$;

b) Normalize $A^i = (A_i - \text{mean}(A))/\text{std}(A)$;

c) Compute weight w based on sr_{hist} (piecewise mapping): $\text{sr} < 0.2 \rightarrow w = 1.7$ (difficult); $0.2 \leqslant \text{sr} < 0.5 \rightarrow w = 1.4$ (medium); $\text{sr} \geqslant 0.5 \rightarrow w = 1.0$ (easy);

d) Weight only positive advantages of correct samples: if $A^i > 0$ and the sample is correct, then $A_i' = w \cdot A^i$; otherwise $A_i' = A^i$.

4) Rationale: With fixed rewards, we perform difficulty-aware scaling on advantages to shift the learning focus toward learnable samples near the competence frontier, alleviating advantage estimation bias under extreme distributions.

Algorithm 1. DB-AW: difficulty-aware advantage weighting component

Require: prompt x, group responses $\{y_i\}_{i=1}^{G}$, rewards $\{r_i\}_{i=1}^{G}$, historical success rate $\text{sr}_{\text{hist}}[x]_i$
Ensure: weighted advantages $\{A_i\}_{i=1}^{G}$
$\text{sr} < -\text{sr}_{\text{hist}}[x]$
return $\{A_i\}_{i=1}^{G}$
// Compute within-group advantages and normalization
baseline ← mean ($\{r_i\}$), $\mu_A \leftarrow 0$, $\sigma_A \leftarrow 1$
for $i = 1$ to G **do**
$A_i \leftarrow r_i -$ baseline
end for
$\mu_A \leftarrow$ mean ($\{A_i\}$), $\sigma_A \leftarrow$ std($\{A_i\}$)
for $i = 1$ to G **do**
$A_i \leftarrow (A_i - \mu_A)/\sigma_A$
end for
// Difficulty weight mapping
$\text{sr} \leftarrow \text{sr}_{\text{hist}}[x]$
if $\text{sr} < 0.2$ **then**
$w \leftarrow 1.7$ {difficult samples}
else if $\text{sr} < 0.5$ **then**
$w \leftarrow 1.4$ {medium difficulty}
else
$w \leftarrow 1.0$ {easy samples}
end if
// Weight positive advantages of correct samples
for $i = 1$ to G **do**
if $A_i > 0$ **and** y_i is correct **then**
$A_i' \leftarrow w \cdot A_i$
else
$A_i' \leftarrow A_i$
end if
end for
return $\{A_i'\}_{i=1}^{G}$

5) Filtering scope and distribution shift: Batch filtering is applied only within a single training step at the batch level. For each batch containing n prompts, each prompt forms a group of G responses (totaling $n \times G$ samples). If a prompt's group yields all-0 or all-1 rewards, it is skipped in the current update. This filtering is not persistent: the same prompt can be included in later steps if resampling yields mixed rewards, ensuring that extremely hard prompts are not permanently excluded. KL regularization further mitigates the forgetting of already-solved prompts (note that for $\text{sr} \geqslant 0.5$, we use $w = 1.0$).

6) Effective update rate: We define the effective update rate as the ratio of mixed-reward groups to total groups per training step, averaged across steps. By construction, filtering removes zero-variance groups, ensuring an effective update rate of 100% for DB-AW, while baseline GRPO can be substantially lower under sparse rewards.

3.3 DB-AW Batch Filtering Component

1) Motivation: Under group sampling, if within-group rewards are all 0 (all incorrect) or all 1 (all correct), gradient signals approach zero, wasting resources and reducing convergence speed. We adopt lightweight group-level filtering to address this.

2) Method: For G responses per prompt, if $\{r_i\}$ is all 0 or all 1, it is identified as a zero-gradient group and removed, retaining only non-zero variance groups for training. Filtering is executed after determining token-level rewards but before advantage estimation, allowing subsequent statistics to inherit non-zero gradient signals.

3) Rationale: By removing zero-variance groups, we ensure that each update contains non-zero gradient signals, avoiding ineffective computation and improving training efficiency.

Algorithm 2. DB-AW batch filtering component

Require: batch $B = \{(x_j, \{r_j, 1, r_j, 2, \cdots, r_j, G\})\}_j|_{=}^{B}|$
Ensure: filtered batch B'
$B' \leftarrow \varnothing$
for each prompt x_j in B **do**
rewards ← $\{r_j, 1, r_j, 2, \cdots, r_j, \text{G}\}$
if all (rewards == 0) **or** all (rewards == 1) **then**
continue {Skip zero-variance groups}
else

7: $B' \leftarrow B' \cup \{(x_j, \text{rewards})\}$ {Retain effective groups}
8: **end if**
9: **end for**
10: **return** B'

4 Experiments

We validate DB-AW's effectiveness on HotpotQA, NQ, and 2WikiMultihopQA. This section details the setup and comparison methods, presents the main results, including cross-model generalization and algorithm comparison, and provides an ablation study on component contributions and synergistic effects.

4.1 Experimental Setup

Datasets: HotpotQA (complex multi-hop reasoning), Natural Questions (open-domain QA), and 2WikiMultihopQA (structured multi-hop reasoning).

Research positioning: Considering the high computational cost of Agentic RAG RL training (where each round requires multiple group responses and search engine interactions), we conduct a controlled comparison on small-to-medium datasets (500/2 000 samples). Our research focuses on validating the proposed algorithmic mechanism's improvement relative to the baseline under the same resource constraints, not on pursuing absolute performance on large datasets.

Controlled experimental principles: All comparison methods use exactly the same datasets, training steps, and hyperparameter configurations, with fixed random seeds and multiple runs averaged. The main metric is Accuracy (strict EM), with unified decoding hyperparameters and retrieval budget.

4.2 Baselines and Variants

GRPO serves as the baseline. DAPO integrates four core technologies. DB-AW (Filtering Only) uses only batch filtering. DB-AW (Weighting Only) uses only difficulty-aware weighting. DB-AW (Full) has both components enabled.

4.3 Implementation Details

Models: Qwen2.5-3B-Instruct (main), Qwen2.5-7B-Instruct (larger version), and LLaMA3.2-3B-Instruct (different architecture). Configuration: n_agent=4 – 6, group size G = 4, and max turns = 3 – 4.

Training hyperparameters: lr = 1e-6, warmup ratio = 0.2 – 0.5, and Kullback-Leibler (KL) coefficient = 0.003 – 0.005.

Rewards and Evaluation: EM is used as the sole outcome-based reward. The main metric is Accuracy (strict EM).

4.4 Main Results

To validate DB-AW's effectiveness relative to mainstream RL methods, we compare the GRPO baseline, DAPO, and DB-AW (full) using the same dataset size and training steps across three models.

Algorithm effectiveness: DB-AW (Full) significantly outperforms GRPO and DAPO across all models and datasets. It achieves the highest relative improvement on Qwen2.5-7B (+18.2%), followed by Qwen2.5-3B (+17.2%) and LLaMA3.2-3B (+15.2%). In contrast, DAPO shows limited improvement (2.7% – 3.4%).

Cross-model generalization: DB-AW achieves significant improvements across all three models (15% – 18%), demonstrating robust generalization across varying parameter scales and model architectures.

DAPO applicability: DAPO's limited improvement suggests that its overlength penalty may hinder valid long-chain reasoning in Agentic RAG. In contrast, DB-AW focuses on two core components without task-specific penalties, thereby maintaining lightweight, general applicability while achieving superior performance.

4.5 Ablation Study

To evaluate component contributions and synergistic effects, we conduct an ablation study within the GRPO framework on Qwen2.5-3B-Instruct.

Accuracy Improvement: Weighting alone achieves a 13.8% improvement, filtering contributes +5.9%, and the full combination reaches +17.2%, indicating that the com-

Table 1. Main results: cross-model generalization (2 000 samples, validation accuracy)

Base Model	Method	HotpotQA	NQ	2Wiki	Avg	Improvement over GRPO
Qwen 2.5-3B	GRPO	0.23	0.49	0.15	0.290	-
	DAPO	0.25	0.49	0.16	0.300	+3.4%
	DB-AW (Full)	**0.31**	**0.53**	**0.19**	**0.340**	**+17.2%**
Qwen 2.5-7B	GRPO	0.25	0.52	0.17	0.313	-
	DAPO	0.27	0.52	0.18	0.323	+3.2%
	DB-AW (Full)	**0.33**	**0.57**	**0.21**	**0.370**	**+18.2%**
LLaMA 3.2-3B	GRPO	0.21	0.45	0.13	0.263	-
	DAPO	0.22	0.45	0.14	0.270	+2.7%
	DB-AW (Full)	**0.26**	**0.49**	**0.16**	**0.303**	**+15.2%**

DAPO: Decoupled Clip and Dynamic Sampling Policy Optimization
DB-AW: dynamic batch-aware advantage weighting
GRPO: group relative policy optimization

Table 2. Ablation study on Qwen2.5-3B-Instruct (2 000 samples, validation accuracy)

Configuration	HotpotQA	NQ	2Wiki	Avg
GRPO (baseline)	0.23	0.49	0.15	0.290
DB-AW (Filtering Only)	0.26	0.50	0.16	0.307 (+5.9%)
DB-AW (Weighting Only)	0.29	0.52	0.18	0.330 (+13.8%)
DB-AW (Full)	**0.31**	**0.53**	**0.19**	**0.340 (+17.2%)**

DB-AW: dynamic batch-aware advantage weighting

bined method outperforms either single component.

Training efficiency: Filtering significantly improves effective update steps by raising the non-zero gradient group ratio from 68% to 100%. Weighting directs focus toward learnable yet challenging samples through difficulty-aware amplification.

Component synergy: The full configuration (+17.2%) significantly exceeds the expected simple addition (≈+14%), indicating that batch filtering provides cleaner gradient signals for difficulty-aware weighting, while weighting enables more effective utilization of filtered samples, forming a virtuous cycle.

5 Conclusions

Addressing Agentic RAG RL training challenges in ACN, we propose DB-AW. Within the GRPO framework, DB-AW introduces two core components: difficulty-aware advantage weighting and batch filtering, which significantly reduce zero-gradient updates and increase the number of non-zero gradient updates per computation unit. Experiments show DB-AW achieves relative improvements of approximately 18%, 17%, and 15% on Qwen2.5-7B, Qwen2.5-3B, and LLaMA3.2-3B, respectively, validating its effectiveness and cross-model generalization.

DB-AW provides a practical pathway for efficient Agentic RAG training in ACN. Experiments show that DB-AW improves the effective update rate from 68% to 100% while maintaining performance, significantly reducing ineffective agent-environment interactions. This has important implications for scalable agent deployment in ACN:

1) Reduced training costs: Fewer ineffective interactions lead to less computational consumption, facilitating massively parallel agent training in ACN. In scenarios where ACN needs large-scale agent deployment, DB-AW's efficiency significantly reduces overall training overhead.

2) Accelerated agent evolution: Efficient training enables agents to quickly adapt to ACN's diverse task scenarios, improving the network's overall intelligence. By directing learning toward the competence frontier, DB-AW helps agents more rapidly enhance generalization on complex reasoning tasks.

3) Support for distributed extension: DB-AW's lightweight and reusable properties lay the foundation for multi-agent collaborative training extension. Batch filtering and difficulty-aware weighting can be conveniently migrated to distributed scenarios, supporting multi-agent knowledge sharing and experience transfer.

Future work is outlined as follows:

1) Multi-agent collaborative training extension: We will extend DB-AW to multi-agent collaborative training in ACN, investigating knowledge sharing and experience transfer mechanisms and exploring efficient gradient aggregation in distributed environments.

2) Larger-scale validation: While we have validated DB-AW on small-to-medium datasets (500/2 000 samples), future work can evaluate scalability and transferability on larger datasets (e.g., the full HotpotQA) and more ACN applications (e.g., intelligent customer service and network operations).

3) Dense reward design exploration: This paper focuses on simple EM-based outcome reward, without studying structured dense rewards such as format scores and process rewards. Future work should systematically explore the role of dense rewards, design principles, and their compatibility with advantage ranking in Agentic RAG RL.

References

[1] China Mobile. AI-Agent communication network white paper [R]. 2025
[2] Singh A, Ehtesham A, Kumar S, et al. Agentic retrieval-augmented generation: a survey on agentic RAG [PP/OL]. arXiv (2025-04-01) [2026-02-06]. https://arxiv.org/abs/2501.09136
[3] Yang Z L, Qi P, Zhang S Z, et al. HotpotQA: a dataset for diverse, explainable multi-hop question answering [PP/OL]. arXiv (2018-09-25) [2026-02-06]. https://arxiv.org/abs/1809.09600
[4] Ho X, Nguyen A K, Sugawara S, et al. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps [C]//The 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, 2020: 6609 – 6625. DOI: 10.18653/v1/2020.coling-main.580
[5] Trivedi H, Balasubramanian N, Khot T, et al. MuSiQue: multihop questions via single-hop question composition [J]. Transactions of the association for computational linguistics, 2022, 10: 539 – 554. DOI: 10.1162/tacl_a_00475
[6] Kwiatkowski T, Palomaki J, Redfield O, et al. Natural questions: a benchmark for question answering research [J]. Transactions of the association for computational linguistics, 2019, 7: 453 – 466. DOI: 10.1162/tacl_a_00276
[7] Yao S Y, Zhao J, Yu D, et al. ReAct: synergizing reasoning and acting in language models [PP/OL]. arXiv (2022-10-06) [2026-02-06]. https://arxiv.org/abs/2210.03629
[8] Asai A, Wu Z Q, Wang Y Z, et al. Self-RAG: learning to retrieve, generate, and critique through self- reflection [PP/OL]. arXiv (2023-10-17) [2026-02-06]. https://arxiv.org/abs/2310.11511
[9] Shinn N, Cassano F, Berman E, et al. Reflexion: language agents with verbal reinforcement learning [PP/OL]. arXiv (2023-03-20) [2026-02-06]. https://arxiv.org/abs/2303.11366
[10] Li X X, Dong G T, Jin J J, et al. Search-o1: agentic search-enhanced large reasoning models [PP/OL]. arXiv (2025-01-09) [2026-02-06]. https://arxiv.org/abs/2501.05366
[11] Shao Z H, Wang P Y, Zhu Q H, et al. DeepSeekMath: pushing the limits of mathematical reasoning in open language models [PP/OL]. arXiv (2024-02-05) [2026-02-06]. https://arxiv.org/abs/2402.03300
[12] Jin B W, Zeng H S, Yue Z R, et al. Search-R1: training LLMs to reason and leverage search engines with reinforcement learning [PP/OL]. arXiv (2025-03-12) [2026-02-06]. https://arxiv.org/abs/2503.09516
[13] Song H T, Jiang J H, Min Y Q, et al. R1-searcher: a novel two-stage outcome-based RL approach for search-enhanced large reasoning models [PP/OL]. arXiv (2025-03-12) [2026-02-06]. https://arxiv. org/abs/2503.05592
[14] Song Y, Ramaneti K, Sheikh Z, et al. Agent data protocol: unifying datasets for diverse, effective fine-tuning of LLM agents [PP/OL]. arXiv (2025-10-28) [2026-02-06]. https://arxiv.org/abs/2510.24702
[15] Eldeeb E, Alves H. Offline multi-agent reinforcement learning for 6G communications: fundamentals, applications and future directions [PP/

OL]. arXiv (2026-01-01) [2026-02-06]. https://arxiv. org/html/2601.00321v1
[16] Christianos F, Papoudakis G, Rahman M A, et al. Scaling multi-agent reinforcement learning with selective parameter sharing [C]//The 38th International Conference on Machine Learning. PMLR, 2021: 1989 – 1998
[17] Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard example mining [C]//Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 761 – 769. DOI: 10.1109/cvpr.2016.89
[18] Soviany P, Ionescu R T, Rota P, et al. Curriculum learning: a survey [J]. International journal of computer vision, 2022, 130(6): 1526 – 1565. DOI: 10.1007/s11263-022-01611-x
[19] Yu Q Y, Zhang Z, Zhu R F, et al. DAPO: an open-source LLM reinforcement learning system at scale [PP/OL]. arXiv (2025-03-18) [2026-02-06]. https://arxiv.org/abs/2503.14476
[20] Ahmadian A, Cremer C, Gallé M, et al. Back to basics: revisiting reinforce style optimization for learning from human feedback in LLMs [PP/OL]. arXiv (2024-02-26) [2026-02-06]. https://arxiv. org/abs/2402.14740
[21] Hu J, Liu J K, Xu H T, et al. REINFORCE++: an efficient RLHF algorithm with robustness to both prompt and reward models [PP/OL]. arXiv (2025-11-10) [2026-02-06]. https://arxiv.org/abs/2501.03262
[22] Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: your language model is secretly a reward model [PP/OL]. arXiv (2024-07-29) [2026-02-06]. https://arxiv.org/abs/2305.18290
[23] Meng Y, Xia M Z, Chen D Q. SimPO: simple preference optimization with a reference-free reward [PP/OL]. arXiv (2024-05-23) [2026-02-06]. https://arxiv.org/abs/2405.14734
[24] Schulman J, Wolskie F, Dhariwal P, et al. Proximal policy optimization algorithms [PP/OL]. arXiv (2017-07-20) [2026-02-06]. https://arxiv.org/abs/1707.06347
[25] Cui G Q, Yuan L F, Wang Z F, et al. Process reinforcement through implicit rewards [PP/OL]. arXiv (2025-02-03) [2026-02-06]. https://arxiv.org/abs/2502.01456
[26] Zhang E C, Yan X G, Lin W, et al. Learning like humans: advancing LLM reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation [C]//The Conference on Empirical Methods in Natural Language Processing. ACL, 2025: 6630 – 6644. DOI: 10.18653/v1/2025.emnlp-main.336

Biographies

Chen Yu received his ME degree in computer science from Southeast University, China in 2017. He currently serves as an AI algorithm engineer at the AI Lab of AsiaInfo Technologies. He holds 3 patents. His research interests include LLMs, reinforcement learning for agents, RAG, multi-agent systems, and the application of AI in telecommunications and enterprise intelligence.

Li Fan is a senior engineer at China Unicom Beijing Branch. He received his ME degree from Beijing University of Posts and Telecommunications. His research interests include 4G/5G network optimization, mobile network digital operation, intelligentization of mobile communication networks, intelligent optimization and operation and maintenance, and emerging mobile communication technologies.

Wu Jie received her BE degree from South China Normal University, China. She currently serves as an AI project expert in the Digitalization Department of China United Network Communications Group Co., Ltd., Guangdong Branch. She holds one patent and multiple software copyrights. Her research interests focus on cutting-edge technologies in artificial intelligence including machine learning, natural language processing (NLP), and multi-modal agents, as well as the innovative application and project management of AI.

Gao Weipeng (gaowp@asiainfo.com) received his ME degree from Jiangnan University, China. He currently serves as an algorithm engineer at the AI Lab of AsiaInfo Technologies. His research interests focus on LLM fine-tuning, time series forecasting, intelligent root cause analysis, agent communication protocols, and intelligent system modeling and optimization.

Ouyang Ye is the Chief Executive Officer and Chief Technology Officer at AsiaInfo Technologies, Co., Ltd. He received his BE degree from Southeast University, China, MS degrees from Tufts University and Columbia University, USA, and his PhD from Stevens Institute of Technology, USA. He has extensive experience in large-scale team management and R&D innovation in the ICT field. He focuses on cross-domain innovation and the commercialization of technologies in cellular networks, AI, and data science. He is also a professor and an IEEE Fellow.