Paper Presentation

FastRL

Learn from the Gap: Differential-Aware Advantage Pruning with Adaptive Rollout Sampling for GRPO

Jiahua Yang et al.

Jinan University · OPPO AI Center

GRPO Saves Memory, but Training Remains Costly

Expensive Rollouts

Multiple responses per question
make rollout generation costly.

Redundant Updates

Token probabilities and gradients
are computed across many trajectories.

Homogeneous Signals

Similar trajectories reinforce repeated reasoning patterns.

Challenge: reduce computation while retaining informative and diverse learning signals.

Not All Trajectories Contribute Equally

Observation: Unequal Contributions

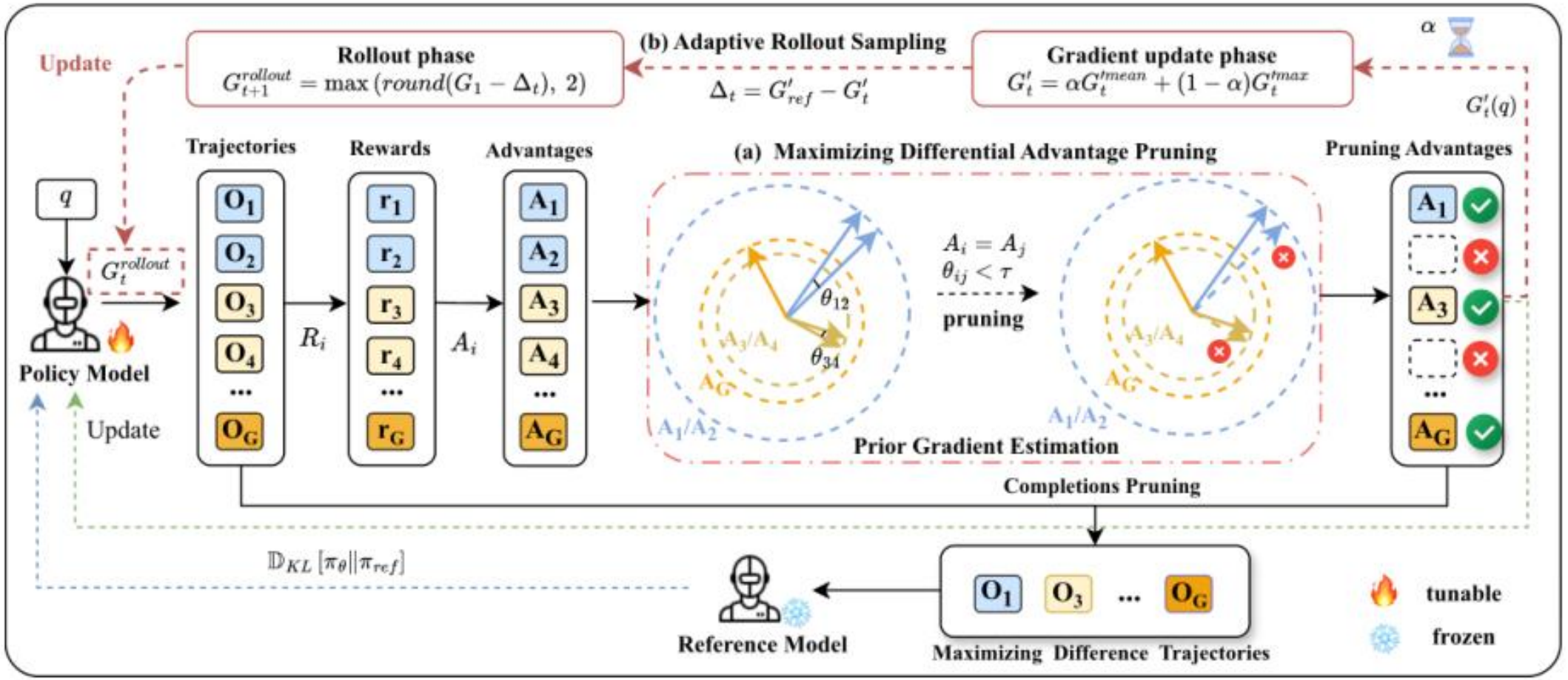
Near-zero advantages give weak update signals.
Similar trajectories with equal advantages may yield redundant gradients.

Idea: Learn from Differences

Select trajectories using advantage groups and structural diversity.
Adapt future sampling budgets to pruning statistics.

Selective updates and adaptive sampling focus computation on informative trajectories.

FastRL: A Feedback Loop of Pruning and Sampling



MDAP: Preserve Diversity Within Advantage Groups

1. Compute Advantages	Score format compliance and answer correctness. Normalize rewards within each question.
2. Group by Advantage	Group trajectories with identical advantages. Compare structural differences within each group.
3. Remove Redundancy	Use token n-gram Jaccard distance as a proxy. Prune if distance to a retained trajectory is below τ.

Retained trajectories are used for policy updates

ARS: Adapt the Rollout Budget to Pruning Statistics

Track Retained Counts

Record the mean and maximum
retained counts at each stage.

Balance Exploration

Favor the maximum count early;
gradually shift toward the mean.

Adjust Future Rollouts

Reduce sampling when redundancy
is high relative to the initial reference.

MDAP reduces update costs, while ARS further reduces generation costs.

Evaluation Across Models and GRPO Variants

Item	Configuration
Training Data	Geometry3K, GeoQA8K-R1V,Math
OOD Benchmarks	MathVerse, MathVision, MathVista, WeMath
Base Models	Qwen2.5-VL-7B, Qwen3-VL-8B, Llama3.1-8B
Baselines	GRPO, DAPO, GSPO, CPPO, GRESO
Metrics	Accuracy (%), training time (h), relative speedup
Implementation	EasyR1; initial rollouts = 8; $\tau = 0.8$; NVIDIA H20

Main Results: Qwen2.5-VL-7B / Geometry3K

Dataset	Method	In-domain	MathVerse	MathVision	MathVista	WeMath	Avg(↑)	Train-Time(↓)	Speed(↑)
Qwen2.5-VL-7B-Instruct									
Geometry3K	GRPO [15]	53.74	43.38	26.25	66.90	68.74	51.32	29.72	1×
	+CPPO [6]	54.41	43.98	27.37	66.30	69.08	51.68	18.52	1.60×
	+GRESO [26]	53.24	42.08	26.09	64.90	67.93	50.25	27.29	1.09×
	+FastRL (Ours)	**55.90**	**45.76**	**27.96**	**67.30**	**69.48**	**52.63**	**13.87**	**2.14×**
	DAPO [22]	53.57	41.24	26.25	67.20	**70.69**	51.35	28.08	1×
	+CPPO	54.57	43.58	27.30	68.40	69.94	52.31	16.61	1.69×
	+GRESO	53.07	43.40	26.68	65.10	68.63	50.95	26.67	1.05×
	+FastRL (Ours)	**54.90**	**43.93**	**27.96**	**69.80**	70.63	**53.08**	**13.29**	**2.11×**
	GSPO [25]	52.41	40.15	26.18	65.40	66.26	49.50	26.33	1×
	+CPPO	53.24	41.27	26.09	**66.60**	69.25	50.80	15.56	1.69×
	+GRESO	52.58	39.92	25.00	65.30	68.10	49.58	25.33	1.04×
	+FastRL (Ours)	**54.57**	**45.48**	**26.97**	66.40	**70.57**	**52.36**	**12.84**	**2.05×**

FastRL delivers 2.05–2.14× speedup and improves OOD Avg across GRPO, DAPO, and GSPO.

Main Results: Qwen2.5-VL-7B / GeoQA8K-R1V

Dataset	Method	In-domain	MathVerse	MathVision	MathVista	WeMath	Avg(↑)	Train-Time(↓)	Speed(↑)
Qwen2.5-VL-7B-Instruct									
GeoQA8K-R1V	GRPO	68.30	45.18	26.78	68.60	68.68	52.31	54.65	1×
	+CPPO	68.03	45.20	26.97	69.00	68.79	52.49	39.22	1.39×
	+GRESO	68.70	**45.99**	26.38	67.20	68.56	52.03	48.33	1.13×
	+FastRL (Ours)	**70.15**	45.91	**27.86**	**70.60**	**70.00**	**53.59**	**25.65**	**2.13×**
	DAPO	67.90	43.96	25.99	68.30	66.67	51.23	53.22	1×
	+CPPO	68.70	44.92	27.43	67.40	67.76	51.88	38.15	1.39×
	+GRESO	67.90	45.05	26.12	64.40	68.10	50.92	46.85	1.14×
	+FastRL (Ours)	**70.29**	**46.17**	**27.53**	**68.50**	**68.97**	**52.79**	**24.62**	**2.16×**
	GSPO	66.97	44.21	27.01	65.60	**68.74**	51.39	52.87	1×
	+CPPO	68.03	45.41	27.63	**67.30**	68.56	52.23	37.55	1.41×
	+GRESO	67.37	42.99	27.07	65.50	66.38	50.49	47.08	1.12×
	+FastRL (Ours)	**69.36**	**46.27**	**27.89**	**67.30**	68.45	**52.48**	**23.85**	**2.22×**

FastRL achieves 2.13–2.22× speedup, with higher OOD Avg than each corresponding baseline.

Main Results: Qwen3-VL-8B / Both Training Sets

Dataset	Method	In-domain	MathVerse	MathVision	MathVista	WeMath	Avg(↑)	Train-Time(↓)	Speed(↑)
Qwen3-VL-8B-Instruct									
Geometry3K	GRPO [15]	78.87	55.76	48.98	71.20	80.63	64.14	30.05	1×
	DAPO [22]	79.53	55.61	50.39	**71.60**	80.06	64.42	28.67	1.05×
	GSPO [25]	79.03	55.28	49.67	71.10	80.69	64.19	28.32	1.06×
	CPPO [6]	79.36	55.03	51.45	70.70	78.79	63.99	22.75	1.32×
	GRESO [26]	78.87	54.82	50.53	70.20	78.22	63.44	29.11	1.03×
	FastRL (Ours)	**80.03**	**56.45**	**54.14**	71.10	**81.61**	**65.83**	**16.37**	**1.84×**
GeoQA8k-R1V	GRPO	82.36	54.01	49.05	70.40	78.22	62.92	85.16	1×
	DAPO	82.36	53.58	50.59	70.00	79.31	63.37	85.01	1.00×
	GSPO	82.49	54.31	49.57	71.50	**79.77**	63.79	83.25	1.02×
	CPPO	84.08	52.64	46.64	69.80	76.00	61.27	68.22	1.25×
	GRESO	82.09	52.74	47.86	69.70	77.01	61.83	84.67	1.01×
	FastRL (Ours)	**84.35**	**55.36**	**52.20**	**71.60**	78.85	**64.50**	**45.33**	**1.88×**

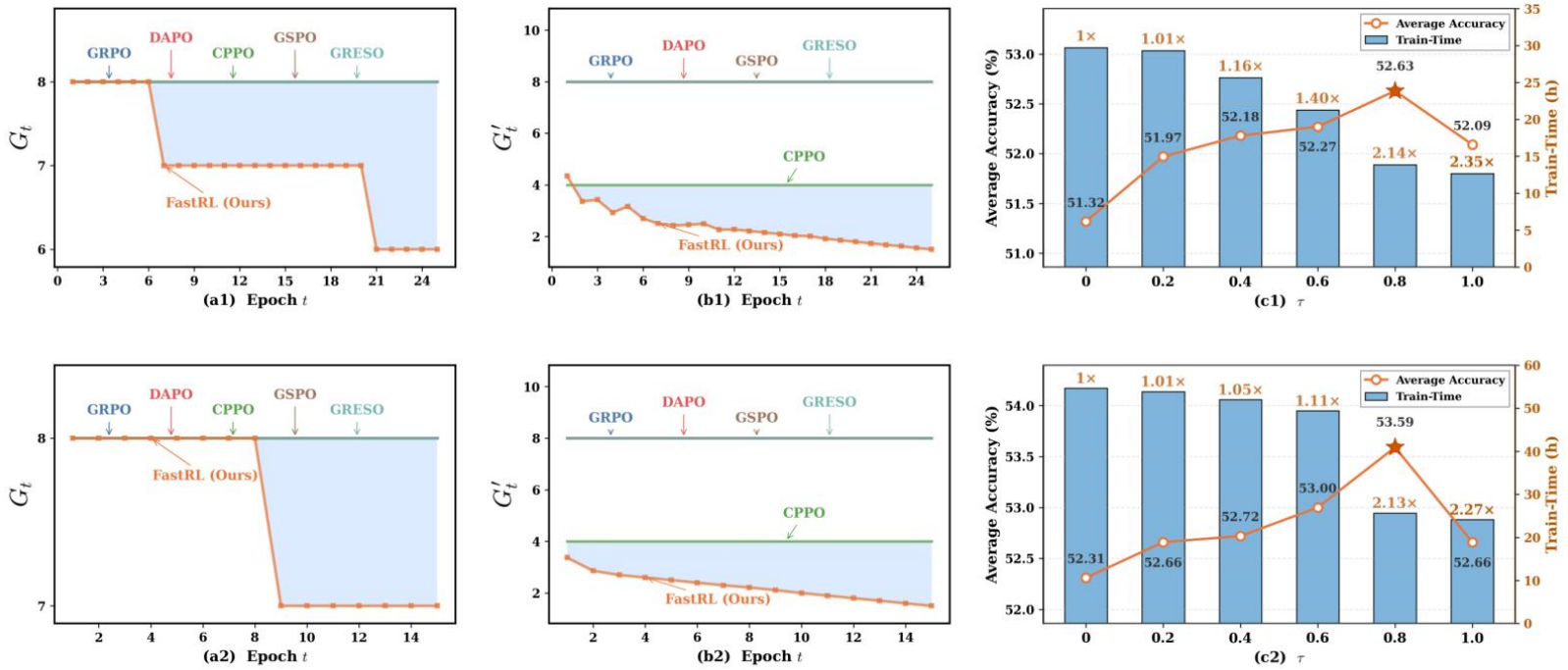
FastRL delivers 1.84× / 1.88× speedup and improves OOD Avg on both training datasets.

Ablation Study: Contributions of MDAP and ARS

Dataset	Method	In-domain	MathVerse	MathVision	MathVista	WeMath	Avg(↑)	Train-Time(↓)	Speed(↑)
Qwen2.5-VL-7B-Instruct									
Geometry3K	FastRL	55.90	45.76	27.96	67.30	69.48	52.63	13.87	2.14×
	w/o MDAP	53.41 ↓2.49	45.00	27.04	66.30	69.31	51.91 ↓0.72	25.85	1.15× ↓0.99×
	w/o ARS	54.01 ↓1.89	45.00	27.99	67.10	68.97	52.27 ↓0.36	14.41	2.06× ↓0.08×
GeoQA8K-R1V	FastRL	70.15	45.91	27.86	70.60	70.00	53.59	25.65	2.13×
	w/o MDAP	68.70 ↓1.45	44.95	27.43	67.60	69.54	52.38 ↓1.21	49.75	1.10× ↓1.03×
	w/o ARS	69.62 ↓0.53	45.38	27.80	70.00	69.83	53.25 ↓0.34	29.53	1.85× ↓0.28×
Qwen3-VL-8B-Instruct									
Geometry3K	FastRL	80.03	56.45	54.14	71.10	81.61	65.83	16.37	1.84×
	w/o MDAP	78.70 ↓1.33	55.56	51.74	70.20	80.11	64.40 ↓1.43	27.42	1.10× ↓0.74×
	w/o ARS	79.70 ↓0.33	56.07	53.85	70.60	80.69	65.30 ↓0.53	17.97	1.67× ↓0.17×
GeoQA8K-R1V	FastRL	84.35	55.36	52.20	71.60	78.85	64.50	45.33	1.88×
	w/o MDAP	82.89 ↓1.46	53.73	51.32	70.10	78.22	63.34 ↓1.16	79.25	1.07× ↓0.81×
	w/o ARS	83.55 ↓0.80	54.92	52.14	71.20	79.14	64.35 ↓0.15	50.12	1.70× ↓0.18×

MDAP drives most speed and accuracy gains; ARS provides additional improvements.

Training Efficiency and Parameter Sensitivity



Rollout and update counts adapt over training; τ = 0.8 balances accuracy and speed.

Generalization to Text-Only Mathematical Reasoning

Dataset	Method	In-domain	AIME2023	AIME2024	AIME2025	AIME2026	Avg(↑)	Train-Time(↓)	Speed(↑)
Llama3.1-8B-Instruct									
MATH	GRPO [15]	72.32	**10.00**	10.00	**6.67**	**3.33**	**7.50**	21.16	1×
	DAPO [22]	72.72	6.67	10.00	**6.67**	**3.33**	6.67	19.72	1.07×
	GSPO [25]	72.56	6.67	10.00	0.00	0.00	4.17	18.89	1.12×
	CPPO [6]	71.92	3.33	10.00	0.00	**3.33**	4.17	11.44	1.85×
	GRESO [26]	71.60	3.33	6.67	3.33	0.00	3.33	17.35	1.22×
	FastRL (Ours)	**73.04**	**10.00**	**13.33**	3.33	**3.33**	**7.50**	**9.22**	**2.30×**

Llama3.1-8B / MATH: 2.30× speedup while matching GRPO's average AIME accuracy.

Exploration Capacity: Pass@16 Evaluation

Dataset	Method	MathVerse	MathVision	MathVista	WeMath	avg@16(↑)
Geometry3K	GRPO [15]	60.53	62.96	82.50	90.34	74.08
	DAPO [22]	58.98	62.57	81.50	90.57	73.41
	GSPO [25]	59.21	62.30	81.90	90.11	73.38
	GRESO [26]	59.44	62.47	82.00	90.23	73.54
	CPPO [6]	60.81	63.19	82.20	90.69	74.22
	FastRL (Ours)	**63.60**	**67.86**	**83.60**	**93.85**	**77.23**
GeoQA8K-R1V	GRPO [15]	60.99	59.41	82.10	89.77	73.07
	DAPO [22]	59.57	57.86	81.90	87.59	71.73
	GSPO [25]	60.53	58.22	82.00	87.64	72.10
	GRESO [26]	59.26	57.24	81.50	86.78	71.20
	CPPO [6]	60.41	58.88	82.00	89.08	72.59
	FastRL (Ours)	**62.39**	**63.95**	**83.20**	**91.32**	**75.21**

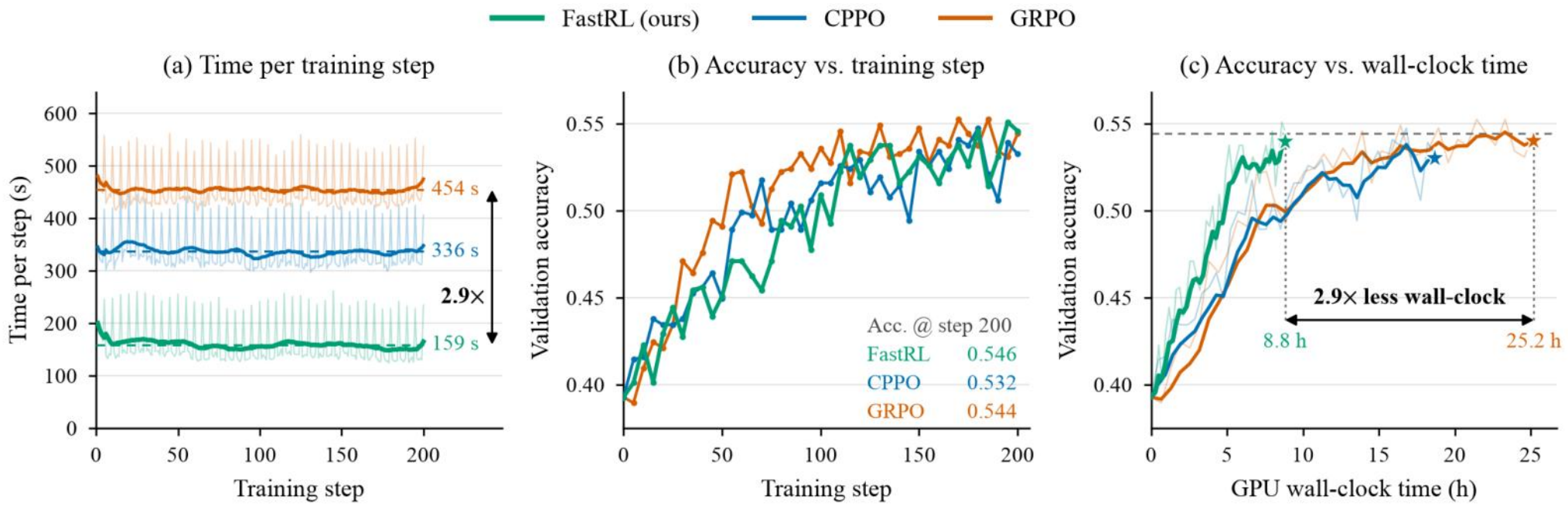
Avg@16 reaches 77.23% / 75.21%, improving over GRPO by 3.15 / 2.14 percentage points.

Pruning Strategies: The Value of Structural Diversity

Pruning Strategy	MathVerse	MathVision	MathVista	WeMath	Avg(↑)	Speed(↑)
Random-pruning 0.25	43.98	27.40	66.90	68.85	51.78	1.27×
Random-pruning 0.50	45.35	26.74	66.60	67.70	51.59	1.60×
Random-pruning 0.75	42.79	26.45	64.60	66.49	50.08	2.10×
Advantage-only (Ours)	44.54	27.57	67.10	69.14	52.09	**2.35**×
FastRL (Ours)	**45.76**	**27.96**	**67.30**	**69.48**	**52.63**	2.14×

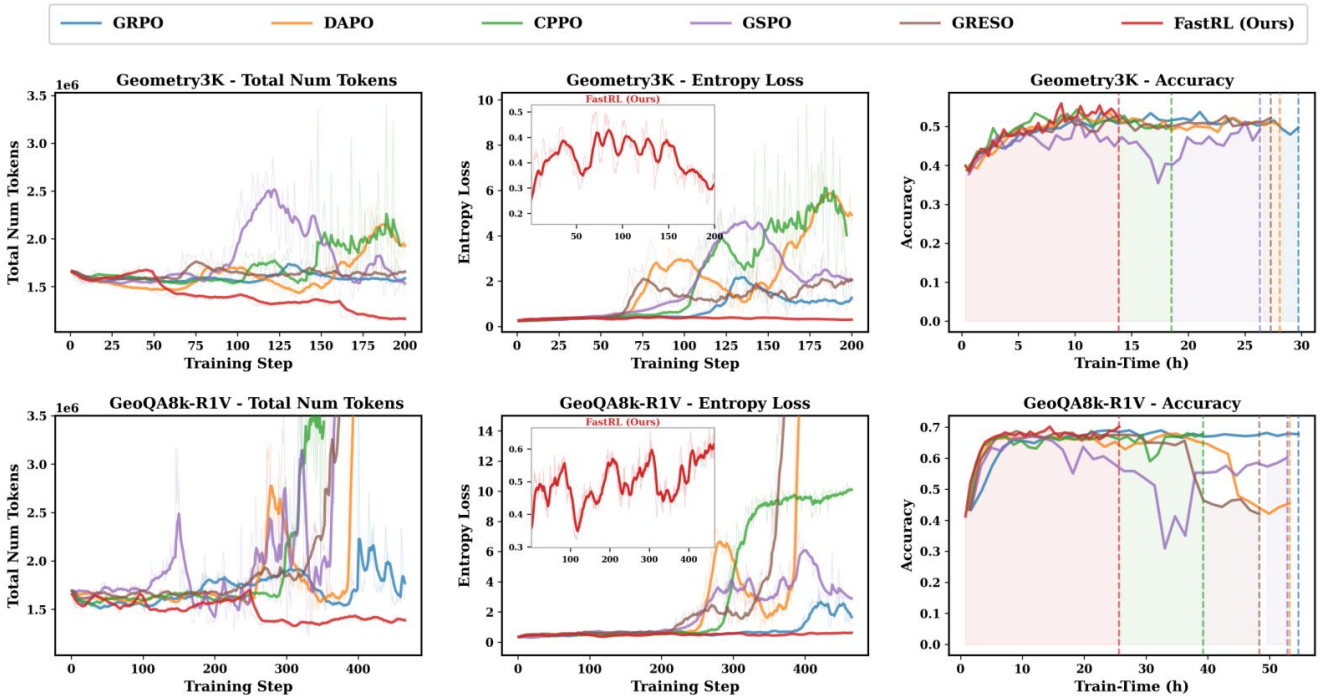
FastRL reaches 52.63% Avg, above Advantage-only (52.09%) and all random-pruning settings.

Low Policy Lag: Efficiency with Aligned Update Counts

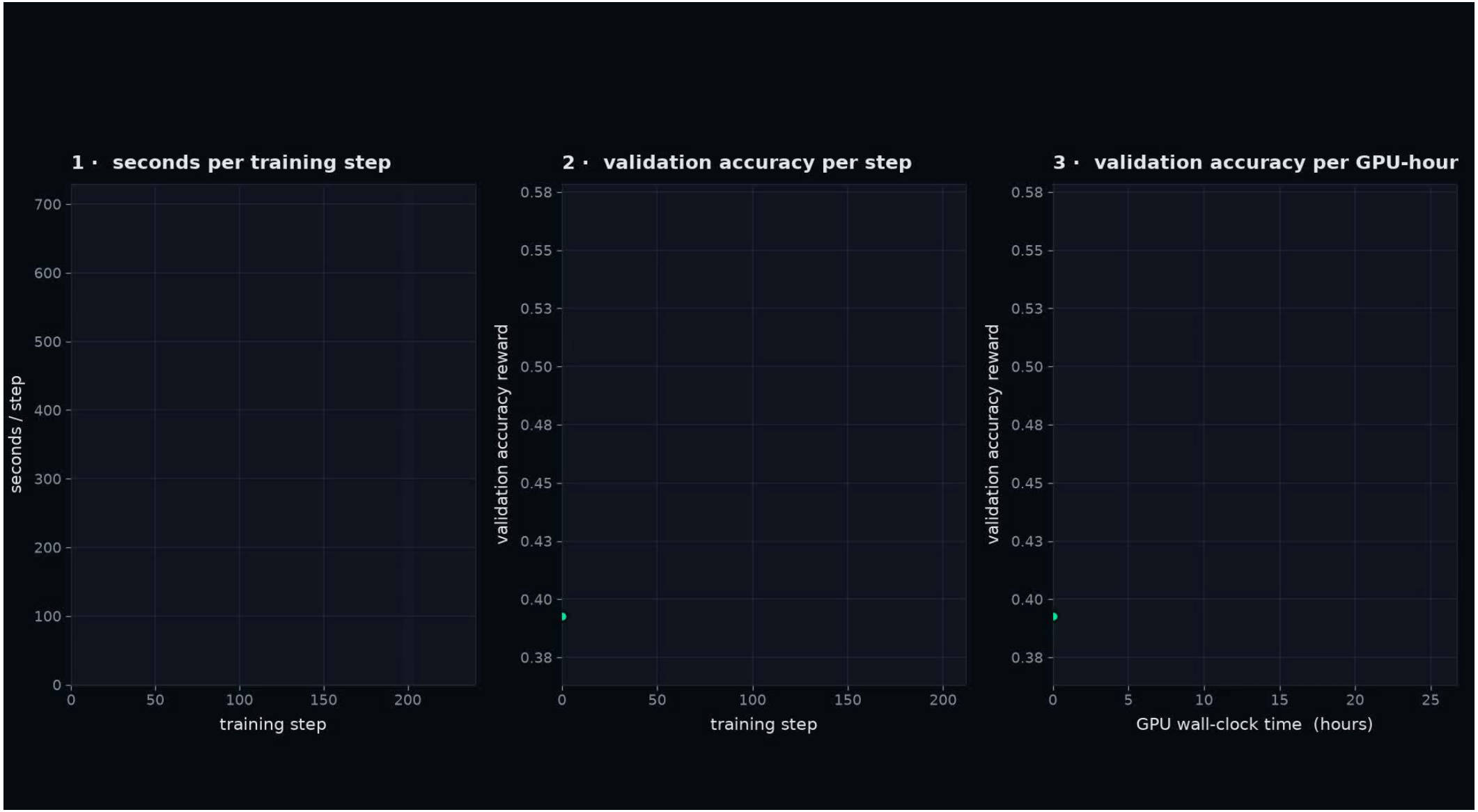


With 4 updates per step, optimized FastRL completes 200 steps in 8.8 h: about 2.9× faster.

High Policy Lag: Cost, Entropy, and Accuracy



With 16 updates per step, FastRL uses fewer tokens, maintains stable entropy, and converges earlier.


1 · seconds per training step
seconds / step
0
100
200
300
400
500
600
700
0
50
100
150
200
training step
2 · validation accuracy per step
validation accuracy reward
0.38
0.40
0.43
0.45
0.48
0.50
0.53
0.55
0.58
0
50
100
150
200
training step
3 · validation accuracy per GPU-hour
validation accuracy reward
0.38
0.40
0.43
0.45
0.48
0.50
0.53
0.55
0.58
0
5
10
15
20
25
GPU wall-clock time (hours)