TreePO Enhancing Policy Efficacy and Inference Efficiency with Tree Modeling

Independent RL rollouts recompute shared prefixes & over-explore. TreePO makes rollout a tree search — same accuracy at up to 43% fewer GPU hours.

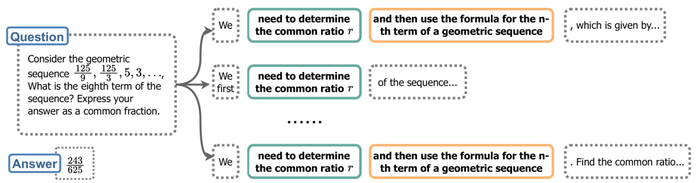
Yizhi Li* Qingshui Gu* Zhoufutu Wen* Ziniu Li* Ruibin Yuan* T. Xing S. Guo T. Zheng X. Zhou X. Qu W. Zhou Z. Zhang W. Shen W. Xue Q. Liu Chenghua Lin◇ Jian Yang◇ Ge Zhang◇ Wenhao Huang◇

*Beihang Univ. · IQuest Research · M-A-P · Seed (ByteDance) · The Univ. of Manchester · HKUST *Equal contribution ◇Corresponding*

ICML 2026 International Conference On Machine Learning

1 The Problem — Rollouts Recompute Shared Prefixes

- **Standard RL (GRPO/DAPO) samples many independent trajectories per query.**
- Shared reasoning prefixes are re-computed every rollout → redundant KV-cache & FLOPs.
- Compute keeps flowing into already-failed paths (no early termination).
- A single sparse outcome reward → coarse, sequence-level credit assignment.



Observation: 16 independent rollouts of one prompt share extensive early/intermediate reasoning segments (matching colors).

2 Our Method — Segment-level Tree Rollout/Sampling

- **Hybrid segment-level tree search + token-level decoding.**
- Branch with budget b = N^d ; transfer budget to active paths (high GPU util).
- Depth-first fallback re-spawns finished paths to reach width w.
- Heuristic early-stop prunes repetitive / low-probability branches — for free.
- Shared prefixes amortized → KV-cache reused across the whole sub-tree.

$$d \times L_{\text{seg}} = B, \qquad b = N^d \text{ branches at depth } d$$

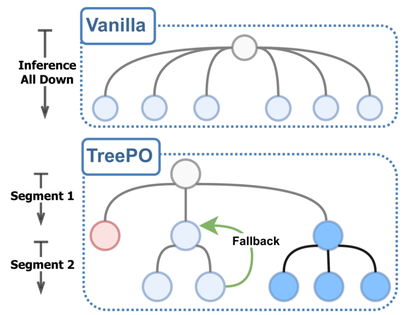
Algorithm · Tree-based Sampling

```
Input: queries Q      Output: rollouts O
P <- Q ;  P <- Branching(P)
while P is not empty:
   S <- Inference(P)          # one segment step
   for each s in S:
      if Finish(s) or Failed(s):
         O <- O + {prefix + s}   # complete leaf
      else:
         P <- P + {prefix + s}   # extend prompt
   P <- Branching(P)          # fork by policy
   P <- Fallback(P, O)        # refill up to w
return O
```

Plug in any early-stop / branching / fallback heuristic — no pipeline bubble.

The Key Idea — Rollout as a Tree Search

Key idea:
- compute the shared prefix once;
- branch only where the model is uncertain;
- prune dead paths & refill to keep GPUs busy.



Vanilla independent rollouts (top) vs. TreePO segment-level tree search (bottom).

1. **Shared prefix** — Computed once → KV-cache reused across the sub-tree.
2. **Dynamic branching** — Fork at uncertainty; budget b = N^d sent to active paths.
3. **Early stop (red)** — Prune repetitive / low-probability "mumbling" paths.
4. **Fallback (green)** — Re-spawn from finished paths to reach width w.

3 Our Method — Tree-based Advantage Estimation

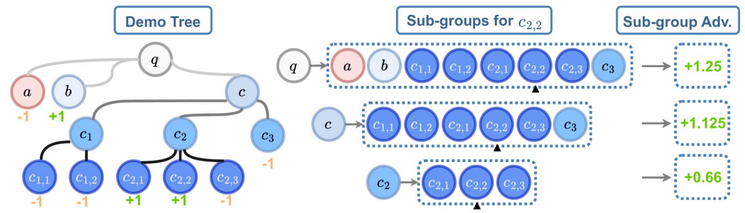
- **Model entire sub-trees as coherent sub-groups (nested by shared predecessor).**
- Aggregate relative advantages across sub-groups + global variance normalization.
- Simple averaging beats size-weighting; root-group term is even redundant.
- Trains directly from a BASE model (RL-zero)

$$G_{|J|} \subseteq G_{|J-1|} \subseteq \cdots \subseteq G_1 \subseteq G$$

$$\widehat{A}_{i,t,j} = R_i - \text{mean}\left(\{R_{i,j}\}^{G_j}\right)$$

$$\widehat{A}_{i,t} = \frac{\sum_{j=1}^{J} \widehat{A}_{i,t,j}}{|J| \cdot \text{std}\left(\{\widehat{A}_{i,t,j}\}^{G}\right)}$$

TreePO advantage: mean-pooled sub-group baselines, globally normalized.



58.21% overall Maj@16 (GRPO 46.63% → +11.6 pts)

−43% GPU hours vs. sequential (12–43% saved overall)

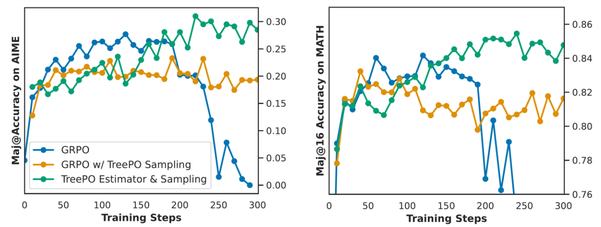
+40% trajectories / sec offline (+30% tokens / sec)

4 It works — 46.6% → 58.2% accuracy

Model	AIME	AMC	MATH	MINERVA	Olympiad	Overall
GRPO	17.13	44.42	72.89	30.94	35.09	46.63
+ TreePO Sampling	19.66	51.63	81.85	33.74	44.76	54.61
TreePO (Fixed Init Div.)	**28.89**	**56.63**	82.41	**35.76**	47.75	56.88
TreePO (More Init Div.)	27.83	55.53	**85.34**	34.98	**49.15**	**58.21**

Maj@16, sequential decoding (peak performance). Bold = best per column. TreePO sampling alone: +8.0 pts; advantage estimator: +2.3–3.6 pts.

Training stability (Maj@16)



AIME (left) & MATH (right): TreePO sampling + advantage (orange/green) trains far more stably than volatile GRPO (blue).

5 The payoff — up to 43% fewer GPU hours

Model	Sampling	Overall	GPU·h
More Init Divergence	Sequential	58.21	6.40
	Tree b=8	58.06	**5.05 ↓22%**
	Tree b=2	54.67	**3.65 ↓43%**
Fixed Init Divergence	Tree b=4	57.50	**4.82 ↓17%**

Contributions

- Tree-based RL rollout: heuristic dynamic divergence + probability fallback, KV-cache reuse.
- Tree-based advantage estimation enabling precise credit assignment from a base model.
- A superior compute–performance trade-off: a more scalable frontier for reasoning RL.

Key takeaways

- **Structure the rollout, don't just sample more — tree search is a free lunch for RL.**
- Sweet spot: depth×segment 14×512; token-aligned segments matter for stable optimization.
- Smaller sub-groups give the most informative credit signal.

Presenter: Yizhi Li
PhD University of Manchester / RS IQuest Research

Project & Code: M-A-P .ai/TreePO