DARA: Few-shot Budget Allocation in Online Advertising via In-Context Decision Making with RL-Finetuned LLMs

Mingxuan Song[1] Yusen Huo[2] Bohan Zhou[1] Shenglin Yin[1] Zhen Xiao[1*] Jieyi Long[3] Zhilin Zhang[2*] Chuan Yu[2]

[1] School of Computer Science, Peking University, Beijing, China [2] Alibaba Group, Beijing, China [3] Theta Labs, Inc., San Jose, USA

1. Problem & Motivation

Personalized budget allocation in online ads is often few-shot / cold-start. Advertisers have individualized goals and rapidly changing environments, so only a small number of historical episodes are available per task.

Key gap. LLMs are strong at *in-context generalization* but weak at *numerically sensitive* structured optimization; RL improves precision but typically needs large interaction data and adapts slowly.

Goal. Allocate a total budget B across T periods:

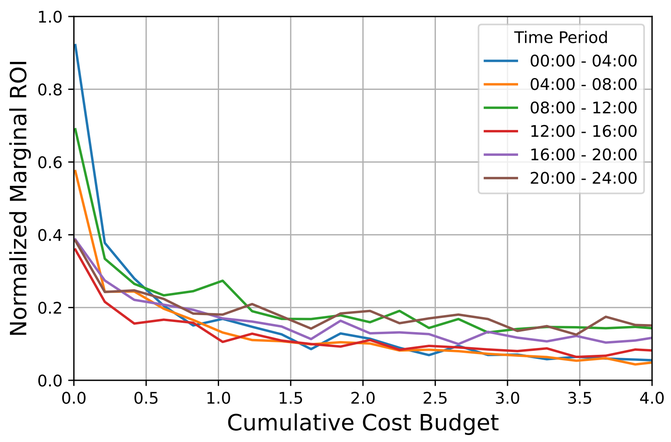
$$\sum_{t=1}^{T} b_t = B, \quad b_t \geq 0,$$

to achieve **balanced marginal ROI** across periods (minimize variance / dispersion of marginal ROI) under few-shot trials.

2. Key Contributions

- **Dual-environment training**: simulation with controllable diminishing-return MROI curves + real-world environment derived from enterprise ad data.
- **DARA**: a *dual-phase* LLM-based decision framework for few-shot budget allocation with interpretability: *Few-shot Reasoner* (early-stage plan) + *Fine-grained Optimizer* (late-stage feedback-driven refinement).
- **GRPO-Adaptive**: RL fine-tuning that *periodically updates the KL reference model* to enable stable yet flexible policy improvement.

3. Environment Modeling



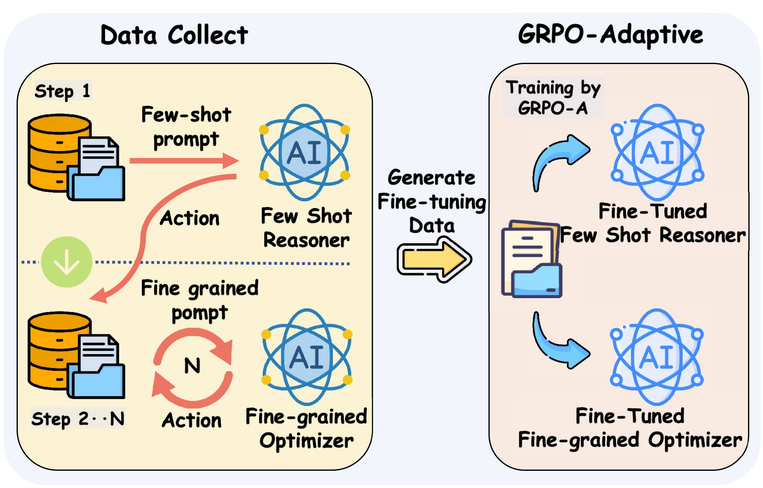
Real-world MROI curves (enterprise ads) + synthetic diminishing-return MROI functions for robust sim-to-real training.

Synthetic MROI. For period i, the marginal ROI decreases with budget until reaching 0:

$$\mathrm{MROI}_i(b) = \max\{F_i(b), 0\}, \quad 0 \leq b \leq B.$$

4. DARA: Dual-Phase Cooperative Agent Architecture

Core idea. Budget allocation is *heterogeneous over time*: early decisions require few-shot generalization; later decisions require numerically sensitive adjustments from feedback.



Dual-phase workflow: Few-shot Reasoner proposes an initial allocation; Fine-grained Optimizer refines with a sliding window of recent feedback.

Agents.

- **Few-shot Reasoner**: consumes few-shot historical episodes and outputs an initial allocation vector (high-level strategy).
- **Fine-grained Optimizer**: takes initial plan + marginal ROI feedback and performs iterative local improvements using a sliding window of recent trials.

5. RL Fine-tuning: GRPO-Adaptive

GRPO-Adaptive objective. Use a GRPO-style clipped advantage objective with KL regularization to a *periodically updated* reference policy:

$$\mathcal{L}_{\text{GRPO-A}}(\theta) = -\mathcal{L}_{\text{adv}}(\theta) + \beta\, \mathbb{D}_{\text{KL}}\left[\pi_\theta \,\|\, \pi_{\text{ref}}\right] \tag{1}$$

$$\mathcal{L}_{\text{adv}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left(\frac{\pi_\theta(o_{i,t}\mid\mathcal{P})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid\mathcal{P})}\hat{A}_{i,t},\right.$$
$$\left.\text{clip}\left(\frac{\pi_\theta(o_{i,t}\mid\mathcal{P})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid\mathcal{P})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right) \tag{2}$$

Why adaptive reference? A fixed KL reference can become outdated as the policy improves, causing over-regularization and hindering later-stage refinement.
where π_{ref} is refreshed every M iterations to track recent policy progress.

Training loop (high level).

1. Sample prompts / environments, generate a group of candidate allocations.
2. Score each candidate by a composite reward; select preferred completions.
3. Update policy with group-wise normalized advantages and clipping.
4. Periodically update $\pi_{\text{ref}} \leftarrow \pi_\theta$.

6. Policy Interface & Reward Design

Action. Allocation vector $b = [b_1, \ldots, b_T]$ extracted from a structured output (e.g., `<answer>[...]</answer>`).
Composite reward.

- **Environment reward** encourages *low dispersion* of marginal ROI across periods:

$$R_{\text{env}}(b) = -\alpha \sum_{i=1}^{T} |r_i - \bar{r}|, \quad r_i = \mathrm{MROI}_i(b_i).$$

- **Constraint penalty** enforces valid dimension and budget sum close to B.
- **Bonus shaping** encourages adaptive refinement relative to the last historical allocation (increase high-reward segments, decrease low-reward segments, with thresholding/clipping).

7. Experimental Setup

Environments.

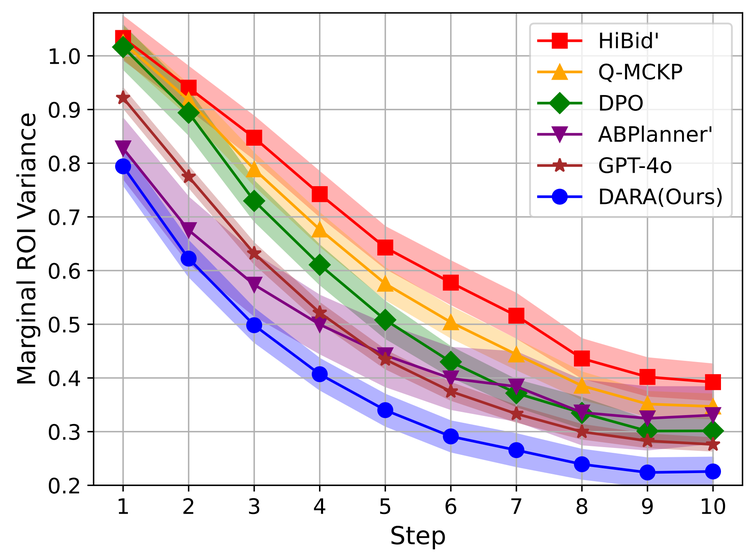
- **Real data environment**: built from online advertising data of a leading global e-commerce platform (high-fidelity simulation).
- **Synthetic environment**: controllable diminishing-return MROI functions for generalization tests.

Metric. Marginal ROI variance measures stability/uniformity of allocation across time periods (lower is better).
Compute. Training conducted on private servers with 8× NVIDIA H20 (96GB) GPUs; 5 runs with 95% confidence intervals.

Alg. 1: DARA: Dual-Phase Cooperative Budget Allocation

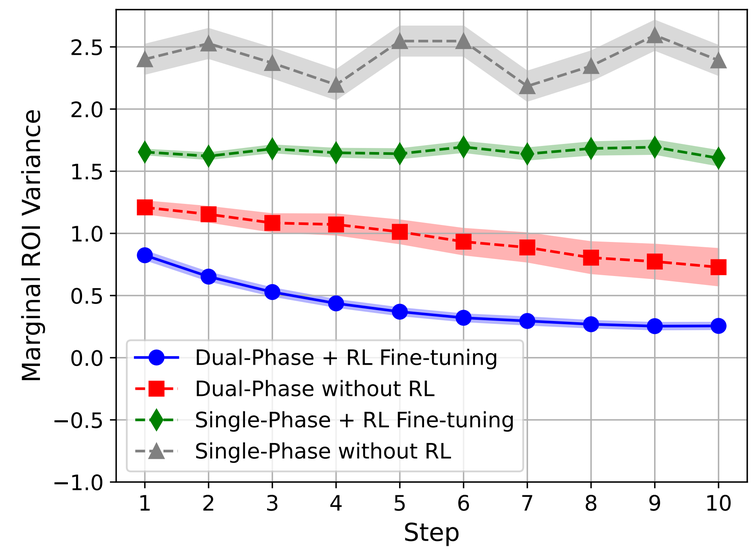
Input: Few-shot historical episodes $\mathcal{H}$, budget constraint B, periods T, sliding window size w
Output: Allocation trajectory $\mathcal{B} = \{b^{(1)}, b^{(2)}, \ldots, b^{(i)}\}$, where $b^{(i)} \in \mathbb{R}^T$
Initialize: $\mathcal{B} \leftarrow [\,]$, episode index $s \leftarrow 1$
Step 1: Few-shot Initialization via Few Shot Reasoner
Generate prompt $\mathcal{P}_1 \leftarrow \text{Encode}(\mathcal{H})$
Initial allocation $b^{(1)} \leftarrow$ Few Shot Reasoner$(\mathcal{P}_1)$
Append $b^{(1)}$ to $\mathcal{B}$
Observe *MROI* feedback $r^{(1)}$
$s \leftarrow s + 1$
Initialize sliding window $\mathcal{W}_2$
Extract the latest $(w-1)$ records from $\mathcal{H}$ as $\mathcal{H}_{\text{tail}}$
$\mathcal{W}_2 \leftarrow \mathcal{H}_{\text{tail}} \cup \{(b^{(1)}, r^{(1)})\}$
Step 2: Fine-Grained Refinement via Fine-grained Optimizer
while True do
  Generate prompt $\mathcal{P}_t \leftarrow \text{Encode}(\mathcal{W}_s)$
  Refined allocation $b^{(s)} \leftarrow$ Fine-grained Optimizer$(\mathcal{P}_t)$
  Append b^s to $\mathcal{B}$
  Observe marginal ROI feedback $r^{(s)}$
  Update sliding window history $\mathcal{W}_{s+1}$ with $(b^{(s)}, r^{(s)})$
  $s \leftarrow s + 1$
  if termination condition met **then break**
return $\mathcal{B}$

8. Main Results

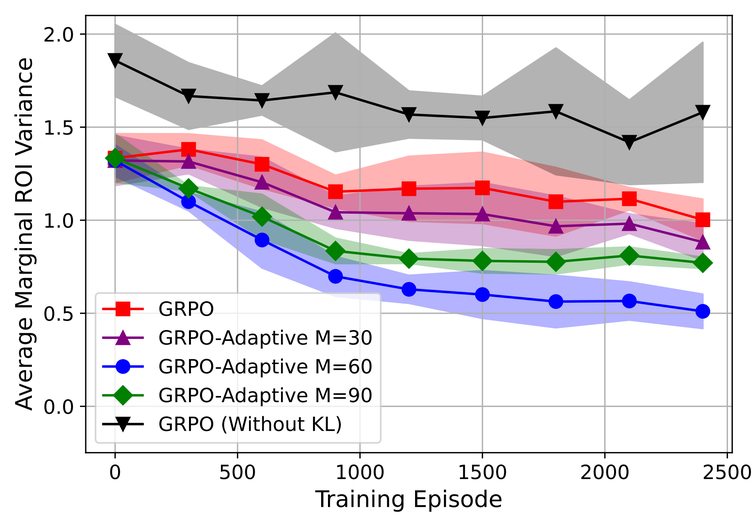


DARA reduces marginal ROI variance across all allocation steps and outperforms baselines (e.g., DPO / RL planners), with larger gains in later steps.

9. Ablations & Scaling



Removing either the Few-shot Reasoner or the Fine-grained Optimizer leads to a substantial increase in marginal ROI variance. The dual-phase architecture enables effective division of global planning and local refinement, which is essential for stable few-shot allocation.



Using a fixed KL reference model causes performance saturation and over-regularization in later training stages. Periodic reference updates in GRPO-Adaptive maintain stability while allowing continual policy improvement.

10. Conclusion & Takeaways

- **Task decomposition matters:** separate early few-shot reasoning from late feedback-driven optimization.
- **RL for LLM decision making:** GRPO-Adaptive stabilizes training while allowing continual improvement via periodic reference updates.
- **Practical impact:** more stable and balanced budget allocation under realistic few-shot constraints.

Links / QR

- Code: `github.com/mx-song/DARA`
- Homepage: `mingxuansong.com`
- Contact: `songmingxuan@pku.edu.cn`

Scan for code