

---

# PGP: Generalized Policy Gradient Theorem for Transformer-based Policies

---

**Hangyu Mao**  
hy.mao@pku.edu.cn

**Guangting Dong**  
dongguanting@ruc.edu.cn

**Zhicheng Dou**  
dou@ruc.edu.cn

## Abstract

We present the **Generalized Policy Gradient (GPG) Theorem**, specifically designed for Transformer-based policies. Notably, we demonstrate that both standard Policy Gradient Theorem and GRPO emerge as special cases within our GPG framework. Furthermore, we explore its practical applications in training Large Language Models (LLMs), offering new insights into efficient policy optimization.

## 1 Introduction

Proximal Policy Optimization (PPO) [24] and Group Relative Policy Optimization (GRPO) [25] rank among the most widely adopted policy gradient algorithms for training Large Language Models (LLMs), which predominantly employ the Transformer architecture [31]. However, these algorithms were initially developed for general reinforcement learning (RL) policies [28] rather than being specifically optimized for Transformer-based policies.

Given the pivotal role of LLMs in AI research, we address a critical question: **Are there policy gradient methods inherently better suited for training Transformer-based LLM policies?** We hypothesize that specialized algorithms could surpass generic policy gradient approaches in both theoretical alignment and empirical performance.

Our primary contribution is a Generalized Policy Gradient (GPG) Theorem tailored for Transformer-based policies. We prove that both the standard Policy Gradient Theorem and GRPO emerge as concrete implementations derived from our GPG framework. Additionally, we investigate practical applications of the GPG Theorem for LLM training, offering insights into its potential advantages over existing methods.

## 2 Preliminary

We consider the standard RL framework [28], in which a learning agent interacts with an environment. The state, action, and reward at each timestep  $t$  are denoted by  $s_t$ ,  $a_t$ , and  $r_t$ , respectively. The environment is characterized by the state transition probabilities  $P(s_{t+1}|s_t, a_t)$  and the reward function  $r_t = R(s_t, a_t)$ . The agent's decision making procedure at each timestep is characterized by a stochastic policy  $\pi_\theta(s_t|a_t) = P(a_t|s_t; \theta) \in [0, 1]$ , with the objective function to maximize the long-term cumulative reward  $R(\tau) = \sum_{t=1}^H r_t$ , where  $\tau = \langle s_1, a_1, r_1, \dots, s_H, a_H, r_H \rangle$  is the decision trajectory and  $H$  is the horizon.

### 2.1 Policy Gradient Theorem

Define the parameterized policy  $\pi_\theta(a_t|s_t)$  and its objective function  $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$ . We use gradient ascent to find the optimal  $\theta^*$  that can maximize the objective:

$$\theta \leftarrow \theta + \alpha * \nabla_\theta J(\theta) \quad (1)$$

The Policy Gradient Theorem [29] says that for any differentiable policy and any objective function, the gradient of the parameterized policy is as follows:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left\{ \sum_{t=1}^H [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) R(\tau)] \right\} \quad (2)$$

A general form of the Policy Gradient Theorem is:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left\{ \sum_{t=1}^H [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \Phi_t] \right\} \quad (3)$$

where  $\Phi_t$  may be one of the following:

- $\sum_{t=1}^H r_t$ : total reward of the trajectory.
- $\sum_{t'=t}^H r_{t'}$ : reward following action  $a_t$ .
- $\sum_{t'=t}^H r_{t'} - b(s_t)$ : baselined version of previous formula.
- $Q^{\pi_{\theta}}(s_t, a_t)$ : state-action value function.
- $A^{\pi_{\theta}}(s_t, a_t)$ : advantage function.
- $r_t + V^{\pi_{\theta}}(s_{t+1}) - V^{\pi_{\theta}}(s_t)$ : Temporal-Difference residual.
- $A^{\text{GAE}(\gamma, \lambda)}(s_t, a_t)$ : the generalized advantage estimator (GAE) for the advantage function.

In practice, the advantage function  $A^{\pi_{\theta}}(s_t, a_t)$  is a common choice because it can achieve better bias-variance trade-off [23].

## 2.2 TRPO, PPO and GRPO

TRPO [22], PPO [24] and GRPO [25] are the special implementations of the Policy Gradient Theorem. PPO and its predecessor TRPO optimize policies (i.e., getting the new policies) with guaranteed monotonic improvement by considering the trust region of old policies. In practice, this is implemented with the off-policy importance sampling strategy (i.e., using old policies  $\pi_{\theta_{\text{old}}}$  to sample trajectories to estimate the gradient of new policies  $\pi_{\theta}$ ):

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \mathbb{E}_{\tau \sim \pi_{\theta}} \left\{ \sum_{t=1}^H [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A^{\pi_{\theta}}(s_t, a_t)] \right\} \\ &= \mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left\{ \sum_{t=1}^H \left[ \frac{\pi_{\theta}(a_t | s_t) \rho_{\theta}(s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t) \rho_{\theta_{\text{old}}}(s_t)} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A_t^{\pi_{\theta}} \right] \right\} \\ &= \mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left\{ \sum_{t=1}^H \left[ \frac{\nabla_{\theta} \pi_{\theta}(a_t | s_t) \rho_{\theta}(s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t) \rho_{\theta_{\text{old}}}(s_t)} A^{\pi_{\theta}}(s_t, a_t) \right] \right\} \\ &\approx \mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left\{ \sum_{t=1}^H \left[ \frac{\nabla_{\theta} \pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} A^{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right] \right\} \end{aligned} \quad (4)$$

where  $\rho_{\theta}(s_t)$  is the station state distribution under policy  $\pi_{\theta}$ . Therefore, the (unclipped) off-policy “surrogate” objective can be represented as:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} \left[ \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} A^{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right] \\ &= \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} [r_t(\theta) A^{\pi_{\theta_{\text{old}}}}(s_t, a_t)] \end{aligned} \quad (5)$$

However, “without a constraint, maximization of the above  $J(\theta)$  would lead to an excessively large policy update” [24], hence, PPO also applies the region clip strategy to penalize changes to the policy that move the ratio  $r_t(\theta)$  away from 1. So the formal (clipped) objective of PPO is:

$$J(\theta) = \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} [\min(r_t(\theta) A^{\pi_{\theta_{\text{old}}}}(s_t, a_t), \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) A^{\pi_{\theta_{\text{old}}}}(s_t, a_t))] \quad (6)$$

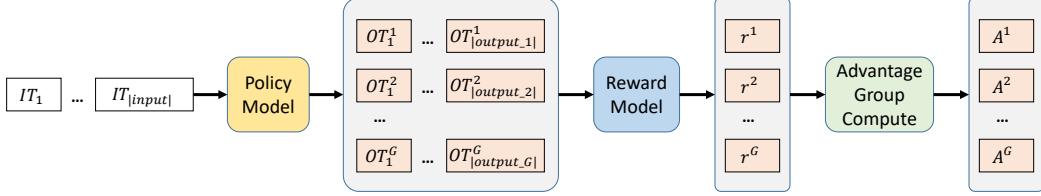


Figure 1: The computing process of GRPO.

where  $\epsilon$  is a small value such as 0.1.

GRPO shares the same objective as PPO, but uses a group of  $G$  output trajectories to compute the advantage  $A^{\pi_{\theta_{\text{old}}}}(s, a)$  as shown in Figure 1. This is especially useful for scenarios (e.g., math and coding) where the partial trajectory is not verifiable, but the whole trajectories can be evaluated with verified rewards.

### 2.3 Chain Rule

The chain rule of probability theory states that for any joint probability distribution  $P(x_1, x_2, \dots, x_n)$ , the following decomposition holds:

$$P(x_1, x_2, \dots, x_n) = P(x_1) \times P(x_2|x_1) \times \dots \times P(x_n|x_1, x_2, \dots, x_{n-1}) \quad (7)$$

Similarly, for any conditional joint probability distribution  $P(x_1, x_2, \dots, x_n|y_1, y_2, \dots, y_m)$ , the chain rule yields the following decomposition:

$$\begin{aligned} P(x_1, x_2, \dots, x_n|y_1, y_2, \dots, y_m) &= P(x_1|y_1, y_2, \dots, y_m) \times \\ &\quad P(x_2|y_1, y_2, \dots, y_m, x_1) \times \\ &\quad P(x_3|y_1, y_2, \dots, y_m, x_1, x_2) \times \dots \\ &\quad P(x_n|y_1, y_2, \dots, y_m, x_1, x_2, \dots, x_{n-1}) \end{aligned} \quad (8)$$

## 3 Generalized Policy Gradient Theorem

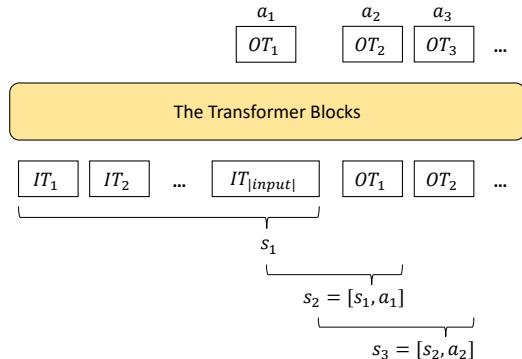


Figure 2: The illustration of Transformer-based polices.

### 3.1 Transformer-based Policy

The Transformer-based policy  $\pi_\theta(a_t|s_t)$ , as illustrated in Figure 2, can be decomposed via the chain rule as follows:

$$\begin{aligned}
& \pi_\theta(OT_1 \mid IT_1, IT_2, \dots, IT_{|\text{input}|}) \times \\
& \pi_\theta(OT_2 \mid IT_1, IT_2, \dots, IT_{|\text{input}|}, OT_1) \times \\
& \pi_\theta(OT_3 \mid IT_1, IT_2, \dots, IT_{|\text{input}|}, OT_1, OT_2) \times \\
& \dots \\
& \pi_\theta(OT_{|\text{output}|} \mid IT_1, \dots, IT_{|\text{input}|}, OT_1, \dots, OT_{|\text{output}|-1}) \\
& = \pi_\theta(OT_1, OT_2, \dots, OT_{|\text{output}|} \mid IT_1, IT_2, \dots, IT_{|\text{input}|}) \\
& = \pi_\theta(MA \mid MS_1)
\end{aligned} \tag{9}$$

where  $IT_i$  and  $OT_i$  are input tokens and output tokens, respectively;  $MS_1 \triangleq \langle IT_1, IT_2, \dots, IT_{|\text{input}|} \rangle$  and  $MA \triangleq \langle OT_1, OT_2, \dots, OT_{|\text{output}|} \rangle$  represent the **macro state** and the **macro action**, respectively.

In general, the output sequence  $\langle OT_1, \dots, OT_{|\text{output}|} \rangle$  can be partitioned into  $K$  segments, yielding generalized macro states and macro actions:

$$\begin{aligned}
MS_i &\triangleq \langle MS_{i-1}, MA_{i-1} \rangle \\
MA_i &\triangleq \langle OT_m, OT_{m+1}, \dots, OT_{m+n} \rangle
\end{aligned} \tag{10}$$

This formulation leads to the following decomposition:

$$\begin{aligned}
& \pi_\theta(MA \mid MS_1) \\
& = \pi_\theta(MA_1 \mid MS_1) \times \\
& \pi_\theta(MA_2 \mid MS_1, MA_1) \times \\
& \dots \\
& \pi_\theta(MA_K \mid MS_1, MA_1, MA_2, \dots, MA_{K-1}) \\
& = \pi_\theta(MA_1 \mid MS_1) \times \\
& \pi_\theta(MA_2 \mid MS_2) \times \\
& \dots \\
& \pi_\theta(MA_K \mid MS_K) \\
& = \prod_{T=1}^K \pi_\theta(MA_T \mid MS_T)
\end{aligned} \tag{11}$$

where  $T$  represents the **macro timestep**.

### 3.2 Derivation of the GPG Theorem

Given the macro states  $MS_i$  and macro actions  $MA_i$  as defined above, we establish the following Generalized Policy Gradient (GPG) Theorem for Transformer-based policies:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left\{ \sum_{T=1}^K [\nabla_\theta \log \pi_\theta(MA_T | MS_T) \Phi_T] \right\} \tag{12}$$

A principal advantage of the GPG Theorem lies in its accommodation of macro-action segments with arbitrary length. This flexible formulation yields significant practical benefits: notably, it naturally supports trajectory segmentation using special tokens (e.g., [SEP], [CLS] or [TOOL]). We elaborate on these applications and implementation considerations in Section 4.

We now present the formal derivation of the Generalized Policy Gradient (GPG) Theorem:

$$\nabla_{\theta} J(\theta) \quad (13)$$

$$= \nabla_{\theta} \mathbb{E}_{\tau \sim \pi_{\theta}} [R(\tau)] \quad (14)$$

$$= \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \quad (15)$$

$$= \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau) \quad (16)$$

$$= \sum_{\tau} P(\tau; \theta) \frac{\nabla_{\theta} P(\tau; \theta)}{P(\tau; \theta)} R(\tau) \quad (17)$$

$$= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} \log P(\tau; \theta) R(\tau) \quad (18)$$

$$= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} [\log \mu(s_1) \prod_{t=1}^H \pi_{\theta}(a_t | s_t) P(s_{t+1} | s_t, a_t)] R(\tau) \quad (19)$$

$$= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} [\log \prod_{t=1}^H \pi_{\theta}(a_t | s_t) P(s_{t+1} | s_t, a_t)] R(\tau) \quad (20)$$

$$= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} [\log \prod_{t=1}^H \pi_{\theta}(a_t | s_t)] R(\tau) \quad (21)$$

$$= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} [\log \prod_{T=1}^K \pi_{\theta}(MA_T | MS_T)] R(\tau) \quad (22)$$

$$= \sum_{\tau} P(\tau; \theta) [\sum_{T=1}^K \nabla_{\theta} \log \pi_{\theta}(MA_T | MS_T)] R(\tau) \quad (23)$$

$$= \sum_{\tau} P(\tau; \theta) [\sum_{T=1}^K \nabla_{\theta} \log \pi_{\theta}(MA_T | MS_T) R(\tau)] \quad (24)$$

$$= \mathbb{E}_{\tau \sim \pi_{\theta}} \{ \sum_{T=1}^K [\nabla_{\theta} \log \pi_{\theta}(MA_T | MS_T) R(\tau)] \} \quad (25)$$

$$= \mathbb{E}_{\tau \sim \pi_{\theta}} \{ \sum_{T=1}^K [\nabla_{\theta} \log \pi_{\theta}(MA_T | MS_T) \Phi_T] \} \quad (26)$$

The key steps in the proof are as follows:

1. The equality between Equations (20) and (21) follows from the deterministic state transition in Transformer-based policies, where:

$$s_{t+1} = [s_t, a_t] \rightarrow P(s_{t+1} | s_t, a_t) = 1 \quad (27)$$

2. The transition from Equation (21) to Equation (22) follows from the autoregressive property of Transformer-based policies, where each state is constructed as  $s_{t+1} = [s_t, a_t]$ . This leads to the

following complete derivation:

$$\begin{aligned}
& \prod_{t=1}^H \pi_\theta(a_t | s_t) \\
& = \pi_\theta(a_1 | s_1) \times \pi_\theta(a_2 | s_2) \times \dots \times \pi_\theta(a_H | s_H) \\
& = \pi_\theta(a_1 | s_1) \times \pi_\theta(a_2 | s_1, a_1) \times \dots \times \pi_\theta(a_H | s_0, a_0, a_1, \dots, a_{H-1}) \\
& = \pi_\theta(a_1, a_2, \dots, a_H | s_1) \\
& = \pi_\theta(MA \mid MS_1) \\
& = \pi_\theta(MA_1 \mid MS_1) \times \\
& \quad \pi_\theta(MA_2 \mid MS_1, MA_1) \times \\
& \quad \dots \\
& \quad \pi_\theta(MA_K \mid MS_1, MA_1, MA_2, \dots, MA_{K-1}) \\
& = \prod_{T=1}^K \pi_\theta(MA_T \mid MS_T)
\end{aligned} \tag{28}$$

3. The transition from Equation (25) to Equation (26) mirrors the generalization in standard Policy Gradient Theorem, i.e., from Equation (2) to Equation (3).

### 3.3 Relation with Existing Methods

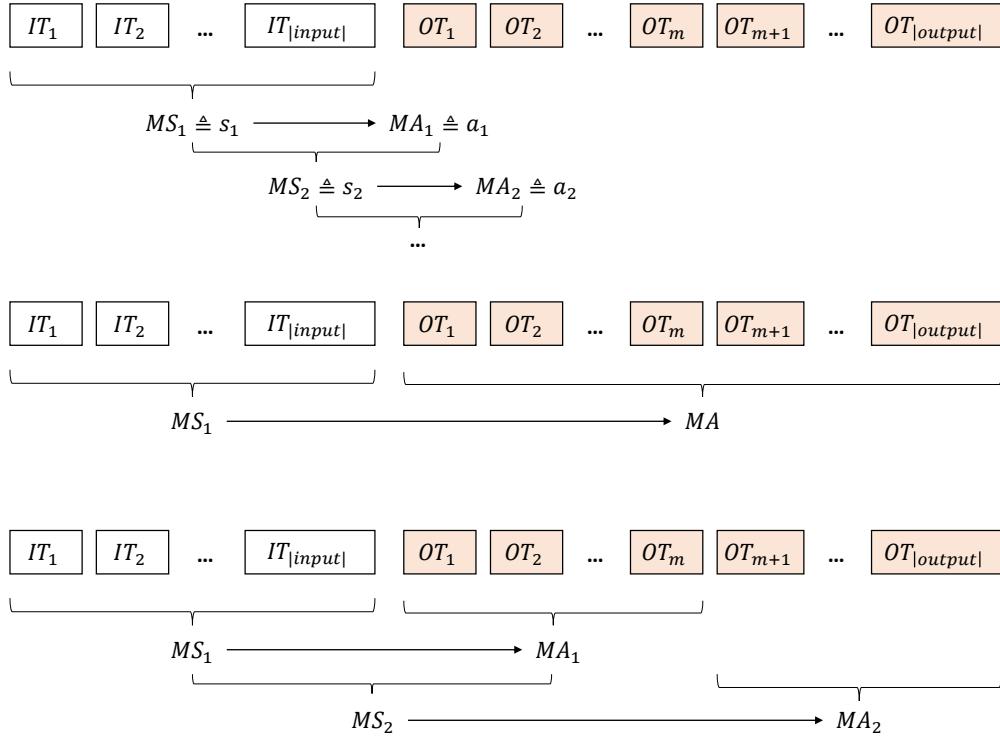


Figure 3: The top shows the special GPG’s policy where  $K = |\text{output}|$ , which is exactly the vanilla Transformer-based policy. The middle shows the special GPG’s policy where  $K = 1$ , which is similar to the GRPO policy. The bottom shows the general form of GPG, and we take  $K = 2$  as an example.

It is straightforward to demonstrate that several existing methods emerge as special cases of our GPG framework, as illustrated in Figure 3. We identify two particularly important cases:

(1) **Token-Level Policy Gradient:** When  $K = |\text{output}|$  (i.e., each macro-action corresponds to a single output token,  $MA_i \triangleq OT_i$ ), our GPG reduces to:

$$\nabla_{\theta} J(\theta) = \mathbb{E} \left[ \sum_{i=1}^{|\text{output}|} \nabla_{\theta} \log \pi_{\theta}(OT_i | MS_i) Q^{\pi}(MS_i, OT_i) \right]$$

which is precisely the standard Policy Gradient Theorem, i.e., Equation (3). This establishes the standard policy gradient as a special case of our generalized framework.

(2) **Sequence-Level Policy Gradient:** When  $K = 1$  (i.e., the entire output sequence comprises a single macro-action), the framework reduces to:  $MA_1 = MA \triangleq \langle OT_1, OT_2, \dots, OT_{|\text{output}|} \rangle$ . This configuration exactly recovers the GRPO paradigm where the complete output sequence functions as an indivisible action unit, reward signals and the advantages are evaluated over batches of complete output sequences, and gradient steps are performed at the full-sequence abstraction level.

## 4 Practical Implementation of GPG Theorem

Figure 4 illustrates the four-phase pipeline for implementing the GPG Theorem in Transformer-based policy optimization: (1) Trajectory Initialization; (2) Macro-action Segmentation; (3) Macro-action Beaming; (4) Advantage Estimation.

### 4.1 Trajectory Initialization

For a given input query, we initialize multiple trajectories using the policy model  $\pi_{\theta}$ , where each trajectory comprises the Transformer’s original token-level outputs. This population-based approach enables exploration of the action space.

### 4.2 Macro-action Segmentation

As previously discussed, the GPG Theorem’s principal advantage lies in its capacity to accommodate macro-action segmentation of arbitrary length. In practical applications, macro-action segmentation can be performed by identifying *marker tokens*. Below, we present three prototypical use cases, while acknowledging that practitioners may adapt this approach to suit their particular requirements.

- **Agentic Reasoning:** Structured agentic tool-using trajectories containing explicit semantic tags (e.g.,  $\langle \text{think} \rangle \dots \langle / \text{think} \rangle$ ,  $\langle \text{tool} \rangle \dots \langle / \text{tool} \rangle$ ) can be segmented at these predefined token boundaries.
- **Document Composition:** In LLM-based text generation, where outputs consist of multiple paragraphs, conventional line breaks (e.g.,  $\backslash n \backslash n$ ) naturally serve as effective segmentation points between compositional units.
- **Creative Problem-Solving:** Unlike routine tasks, creative processes often present greater challenges for LLMs. Here, high-entropy tokens (those exhibiting greater predictive uncertainty) may be employed as segmentation boundaries.

### 4.3 Macro-action Beaming

For each marker token at position  $t$ :

- Macro-state is the partial trajectory before this token, i.e.,  $MS = \langle IT_1, IT_2, \dots, IT_{|\text{input}|}, OT_1, \dots, OT_t \rangle$
- Macro-action is the partial trajectory after this token, i.e.,  $MA = \langle OT_{t+1}, \dots, OT_{|\text{output}|} \rangle$

In the macro-action beaming step, we further generate  $N$  candidate continuations ( $MA^{(1)}, \dots, MA^{(N)}$ ) from each  $MS$ , enabling diverse exploration.

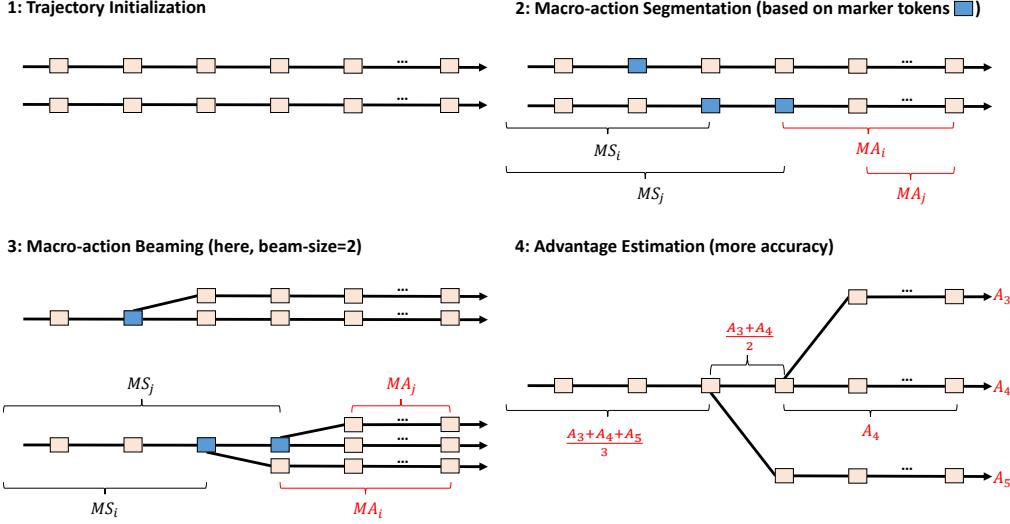


Figure 4: The practical application of the GPG Theorem for LLM-based policy training.

#### 4.4 Advantage Estimation

We propose calibrated advantage computation, which improves upon GRPO’s approach:

- **Reward Computation:** Obtain trajectory rewards  $r_i$  via reward model or rule-based scoring.
- **Initial Advantage:**  $A_i^{init} = \frac{r_i - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}}}$  where  $\mathcal{G}$  is the trajectory group.
- **Token-level Calibration:** For token  $a_t$  in trajectory  $i$ :

$$A_t = \frac{1}{|\mathcal{S}_t|} \sum_{j \in \mathcal{S}_t} A_j^{init} \quad (29)$$

where  $\mathcal{S}_t$  is the set of trajectories sharing prefix  $a_{[1:t-1]}$

#### 4.5 Policy Optimization

Using the calibrated advantages, we can use the classical RL methods like PPO to optimize the Transformer-based policy’s model parameters as shown by Equation (6).

### 5 Experiment

We validate our GPG framework on agentic reasoning tasks - specifically, LLM-based tool-use agent training - selected for two principal advantages: (1) Predefined semantic tags provide deterministic segmentation boundaries, removing the ambiguity of learned marker tokens; (2) The complex action space and delayed rewards in tool manipulation present a rigorous testbed for policy optimization methods. We term our approach Agentic Reinforced Policy Optimization (ARPO) to reflect its specialized application domain <sup>1</sup>.

#### 5.1 Dataset

We evaluate our approach on two challenging categories of long-horizon agentic reasoning tasks.

**Mathematical Reasoning:** GSM8K, MATH [7], MATH500 [17], AIME2024 and AIME2025 <sup>2</sup>.

**Knowledge-Intensive Reasoning:** WebWalker [34], HotpotQA [36], 2WikiMultihopQA [8], Musique [30] and bamboogle [18].

<sup>1</sup>See more details in our paper: <https://arxiv.org/abs/2507.19849>

<sup>2</sup><https://huggingface.co/datasets/AI-M0/aimo-validation-aime>

Method	Mathematical Reasoning					Knowledge-Intensive Reasoning					
	AIME24	AIME25	MATH500	GSM8K	MATH	WebWalker	HQA	2Wiki.	MuSiQ.	Bamb.	Avg.
<b>Qwen2.5-3B</b>	10.0	6.7	63.0	75.0	71.6	0.5	9.7	9.4	3.6	11.7	26.1
+ TIR Prompting	6.7	6.7	52.2	56.6	62.8	14.0	15.4	14.1	6.1	16.4	25.1
+ GRPO	<u>20.0</u>	13.3	<b>72.0</b>	<b>86.0</b>	81.0	<u>21.0</u>	<u>56.5</u>	64.5	24.7	65.2	50.4
+ Reinforce ++	16.7	13.3	70.4	<u>85.0</u>	80.2	19.5	55.9	62.3	27.9	<u>65.7</u>	49.7
+ DAPO	<u>20.0</u>	<u>16.7</u>	71.2	<u>85.0</u>	<u>81.2</u>	19.5	54.8	62.5	<b>30.0</b>	64.8	50.6
+ ARPO	<b>23.3</b>	<b>20.0</b>	<u>71.4</u>	<u>85.0</u>	<b>82.5</b>	<b>24.5</b>	<b>58.5</b>	<b>67.4</b>	<u>28.7</u>	<b>66.8</b>	<b>52.8</b>
<b>Qwen2.5-7B</b>	10.0	10.0	70.6	90.2	82.0	2.0	12.2	12.6	6.6	24.0	32.0
+ TIR Prompting	6.7	10.0	68.2	64.6	78.2	15.5	14.8	18.3	9.5	23.6	31.0
+ GRPO	23.3	<u>26.7</u>	78.0	<b>92.8</b>	<u>87.8</u>	22.0	<b>59.0</b>	<b>76.1</b>	<u>30.6</u>	<b>68.4</b>	<b>56.5</b>
+ Reinforce ++	<u>26.7</u>	23.3	78.0	<u>92.2</u>	<b>88.8</b>	<b>26.0</b>	55.1	<u>68.9</u>	25.2	64.9	54.9
+ DAPO	20.0	23.3	<b>80.4</b>	91.0	<b>88.8</b>	24.0	57.7	68.4	28.6	65.5	54.8
+ ARPO	<b>30.0</b>	<b>30.0</b>	<u>78.8</u>	<u>92.2</u>	<b>88.8</b>	<b>26.0</b>	<u>58.8</u>	<b>76.1</b>	<b>31.1</b>	<b>71.5</b>	<b>58.3</b>
<b>Llama3.1-8B</b>	3.3	0.0	43.3	81.4	60.6	3.0	24.3	24.6	10.4	40.0	28.8
+ TIR Prompting	3.3	3.3	39.4	73.8	58.2	15.0	48.5	47.5	15.5	58.4	36.3
+ GRPO	13.3	<u>13.3</u>	<b>62.4</b>	<u>87.4</u>	<u>79.2</u>	26.5	<u>57.8</u>	<u>71.8</u>	<u>31.0</u>	68.2	<u>51.1</u>
+ Reinforce ++	13.3	<b>16.7</b>	61.4	87.0	77.2	<u>27.5</u>	57.1	71.6	29.9	<u>69.1</u>	<u>51.1</u>
+ DAPO	16.7	<u>13.3</u>	61.2	<u>87.4</u>	76.4	25.5	56.6	70.3	29.2	67.3	50.4
+ ARPO	23.3	<b>16.7</b>	<b>64.6</b>	<b>88.0</b>	<b>80.2</b>	<b>30.5</b>	<b>65.4</b>	<b>75.5</b>	<b>34.8</b>	<b>73.8</b>	<b>55.3</b>

Table 1: The comprehensive evaluation results across 10 challenging reasoning tasks are presented. For clarity, we highlight the top two performing methods using **bold** and underline formatting. The following dataset abbreviations are used: HQA (HotpotQA), 2Wiki. (2wikiMultiHopQA), MuSi. (MuSiQue), and Bamb (Bamboogle).

## 5.2 Baselines.

To comprehensively assess the performance of ARPO, we employ the following three baseline methodologies.

**Direct Reasoning Approaches:** We evaluate instruction-tuned versions of both the Qwen2.5 [19] and Llama3.1 [6] model families.

**Trajectory-level RL Methods:** We conduct comparative evaluations between ARPO and established trajectory-level RL algorithms commonly employed for training LLM-based tool-use agents, including GRPO [26], DAPO [37], and the REINFORCE++ [9].

## 5.3 Evaluation Protocol

To ensure consistency with established reasoning benchmarks, we employ a browser-enabled search engine and a python code interpreter as tools for evaluation. For knowledge-intensive reasoning tasks, we measure accuracy using F1 scores. All other tasks are assessed under the LLM-as-Judge paradigm using the Qwen2.5-72B-instruct model.

We implement pass@1 evaluation with stochastic sampling, configuring the temperature parameter to 0.6 and top-p to 0.95, respectively. Following prior methodology [14], we extract model responses by identifying text segments delimited by \box markers in the output. This standardized approach ensures fair and reproducible comparison across all evaluated tasks.

## 5.4 Main Results

The key experimental results are presented in Table 1. Under standardized evaluation conditions, ARPO demonstrates consistent superiority over all trajectory-level RL baselines, with the following critical observations.

**Inherent Limitations of Prompting Approaches:** Our analysis reveals that Tool-integrated prompting (TIR) methods [14] exhibit fundamental constraints in discovering optimal tool-use strategies. Across both Qwen and Llama model families, TIR prompts yield marginal performance gains that frequently underperform direct reasoning baselines. These results indicate that prompt engineering alone cannot effectively guide LLMs toward sophisticated tool manipulation while preserving their native reasoning abilities.

**Challenges in Trajectory-Level Optimization:** The comparative analysis exposes significant limitations in conventional trajectory-level RL algorithms. While DAPO shows competence in single-turn reasoning, its performance degrades markedly in multi-turn tool interaction scenarios, particularly for knowledge-intensive tasks. This empirical evidence corroborates our hypothesis that

trajectory-level approaches are fundamentally constrained in facilitating granular, step-wise tool behavior learning in LLMs.

**Consistent Superiority of ARPO:** ARPO achieves state-of-the-art performance across all 10 benchmark datasets, delivering an average accuracy gain of 4% over competing methods while maintaining robust performance across diverse domains. Notably, the framework demonstrates remarkable backbone-agnostic properties, showing substantial improvements for both Qwen and Llama model series. These results validate ARPO’s effectiveness as a versatile and adaptable solution for tool-augmented language models.

**Model Capacity Scaling:** The results demonstrate consistent performance improvements as model size increases across all methods. Notably, ARPO achieves the most significant gains from scaling, with 7B models showing 5.5% absolute improvement over 3B variants (58.3% vs 52.8%). The performance delta between ARPO and baselines widens with larger models (e.g., +7.8% over DAPO for 7B vs +2.2% for 3B). Llama3.1-8B shows particular sensitivity to optimization methods, with ARPO delivering 4.2% improvement over the next best method.

In summary, these results collectively demonstrate that ARPO’s step-level optimization paradigm fundamentally addresses key limitations in existing approaches for tool-augmented language models. The consistent performance advantages across model sizes and task categories suggest the framework’s robustness and generalizability.

## 6 Related Work

### 6.1 Policy Gradient in Reinforcement Learning

The Policy Gradient Theorem [29] provides the foundational framework for gradient-based policy optimization in reinforcement learning. The simplest form, REINFORCE [33], estimates the gradient as  $\nabla_{\theta} J(\theta) = \mathbb{E}[\nabla_{\theta} \log \pi_{\theta}(a|s)Q^{\pi}(s, a)]$ , where  $\pi_{\theta}$  represents the policy parameterized by  $\theta$ , and  $Q^{\pi}(s, a)$  is the long-term value of taking action  $a$  at state  $s$ . While straightforward, this approach suffers from high variance and inefficient exploration. To address these limitations, several advanced variants have been developed. Natural Policy Gradient [10] incorporates the Fisher information matrix  $F(\theta)$  to enable invariant updates:  $\tilde{\nabla}_{\theta} J(\theta) = F(\theta)^{-1} \nabla_{\theta} J(\theta)$ . This provides more stable convergence by accounting for the underlying geometry of the parameter space. Deterministic Policy Gradient (DPG) [27] extends the framework to deterministic policies  $\mu_{\theta} : \mathcal{S} \rightarrow \mathcal{A}$ , with gradient  $\nabla_{\theta} J(\theta) = \mathbb{E}[\nabla_{\theta} \mu_{\theta}(s) \nabla_a Q^{\mu}(s, a)|_{a=\mu_{\theta}(s)}]$ . This is particularly effective in continuous action spaces. Trust Region Policy Optimization (TRPO) [22] constrains policy updates within a trust region to guarantee monotonic improvement, while Proximal Policy Optimization (PPO) [24] simplifies this through clipped objective functions.

### 6.2 Policy Gradient in LLM Optimization

Recent advances have significantly advanced policy optimization techniques for language tasks, with Reinforcement Learning from Human Feedback (RLHF) and Reinforcement Learning with Verifiable Rewards (RLVR) [11, 13] emerging as dominant paradigms for large language model alignment. These approaches have demonstrated remarkable success in aligning model outputs with human preferences while maintaining generation quality. Building upon these foundations, Generalized Reinforcement Policy Optimization (GRPO) introduces enhanced advantage estimation through grouped trajectory analysis within the PPO framework, offering improved stability in complex language generation tasks. Direct Preference Optimization (DPO) [20] bypasses explicit reward modeling by optimizing policies directly from offline preference datasets, while Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO) [37] exploring algorithm design across various RL modules.

### 6.3 Agentic Policy Optimization

There are different categories of agentic policy optimization. Some focus on prompt optimization for LLM-based agents, e.g., TPTU [21]; some for supervised training LLMs for agentic tool-using, e.g., TPTU-v2 [12]; some for cooperation within multiple LLM-based agents, e.g., LLaMAC [38]; some for structured data like sheet and sql operation [39, 16, 35, 1, 32]; some for unstructured data

retrieval like agentic rag [15, 2]. Recently, using RL to optimize LLMs for agentic tool-using is very popular, e.g., tool-star [4], arpo [5], and aepo [3]; the work in this paper presents a foundation theory for this kind of agentic policy optimization.

#### 6.4 Our Contribution

The existing methodological advances have substantially pushed the boundaries of policy optimization, especially in complex environments characterized by high-dimensional state and action spaces. However, existing approaches remain generic solutions applicable to all policy types, failing to specifically account for the unique auto-regressive nature of Transformer architectures that underpin modern LLMs. This represents a critical limitation, as the sequential decision-making process in Transformers differs fundamentally from conventional reinforcement learning settings.

Our work bridges this gap through two key contributions: First, the proposed GPG Theorem is specifically designed for Transformer-based policies, explicitly modeling their auto-regressive properties. Second, we develop ARPO as an instantiation of the GPG Theorem, providing the first policy optimization framework that natively respects the architectural constraints of LLMs while being specifically tailored for agentic reasoning tasks. This represents a paradigm shift from previous general-purpose policy optimization methods to architecture-aware optimization for LLMs.

## 7 Conclusion

This paper presents the Generalized Policy Gradient (GPG) Theorem, a novel theoretical framework specifically tailored for optimizing Transformer-based policies. Our analysis establishes that both the conventional Policy Gradient Theorem and GRPO can be naturally derived as special cases within our unified GPG framework. Beyond theoretical contributions, we provide practical implementation guidelines for effectively applying GPG to LLM training scenarios. Comprehensive evaluations demonstrate the superior performance of our approach. These findings advance our understanding of efficient policy optimization techniques for modern language models.

## References

- [1] Yibin Chen, Yifu Yuan, Zeyu Zhang, Yan Zheng, Jinyi Liu, Fei Ni, Jianye Hao, Hangyu Mao, and Fuzheng Zhang. Sheetagent: towards a generalist agent for spreadsheet reasoning and manipulation via large language models. In *Proceedings of the ACM on Web Conference 2025*, pages 158–177, 2025.
- [2] Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. *arXiv preprint arXiv:2504.18243*, 2025.
- [3] Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, et al. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*, 2025.
- [4] Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *CoRR*, abs/2505.16410, 2025.
- [5] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.
- [6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [7] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information*

- [8] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics, 2020.
- [9] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.
- [10] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.
- [11] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *Trans. Mach. Learn. Res.*, 2025, 2025.
- [12] Yilun Kong, Jingqing Ruan, Yihong Chen, Bin Zhang, Tianpeng Bao, Shi Shiwei, Du Qing, Xiaoru Hu, Hangyu Mao, Ziyue Li, et al. Tptu-v2: Boosting task planning and tool usage of large language model-based agents in real-world industry systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 371–385, 2024.
- [13] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024.
- [14] 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. *CoRR*, abs/2501.05366, 2025.
- [15] Zhicong Li, Jiahao Wang, Zhishu Jiang, Hangyu Mao, Zhongxia Chen, Jiazen Du, Yuanxing Zhang, Fuzheng Zhang, Di Zhang, and Yong Liu. Dmqr-rag: Diverse multi-query rewriting for rag. *arXiv preprint arXiv:2411.13154*, 2024.
- [16] Zhishuai Li, Xiang Wang, Jingjing Zhao, Sun Yang, Guoqing Du, Xiaoru Hu, Bin Zhang, Yuxiao Ye, Ziyue Li, Rui Zhao, et al. Pet-sql: A prompt-enhanced two-round refinement of text-to-sql with cross-consistency. *arXiv preprint arXiv:2403.09732*, 2024.
- [17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [18] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics, 2023.
- [19] Qwen, : An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024.

- [20] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- [21] Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Hangyu Mao, Ziyue Li, Xingyu Zeng, Rui Zhao, et al. Tptu: Task planning and tool usage of large language model-based ai agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.
- [22] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.
- [23] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.
- [24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [25] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [26] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.
- [27] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.
- [28] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [29] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
- [30] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.
- [31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [32] Zhongyuan Wang, Richong Zhang, Zhijie Nie, and Hangyu Mao. General table question answering via answer-formula joint generation. *arXiv preprint arXiv:2503.12345*, 2025.
- [33] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.
- [34] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal. *CoRR*, abs/2501.07572, 2025.
- [35] Sun Yang, Qiong Su, Zhishuai Li, Ziyue Li, Hangyu Mao, Chenxi Liu, and Rui Zhao. Sql-to-schema enhances schema linking in text-to-sql. In *International Conference on Database and Expert Systems Applications*, pages 139–145. Springer, 2024.
- [36] 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 *EMNLP*, pages 2369–2380, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.

- [37] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- [38] Bin Zhang, Hangyu Mao, Jingqing Ruan, Ying Wen, Yang Li, Shao Zhang, Zhiwei Xu, Dapeng Li, Ziyue Li, Rui Zhao, et al. Controlling large language model-based agents for large-scale decision-making: An actor-critic approach. *arXiv preprint arXiv:2311.13884*, 2023.
- [39] Bin Zhang, Yuxiao Ye, Guoqing Du, Xiaoru Hu, Zhishuai Li, Sun Yang, Chi Harold Liu, Rui Zhao, Ziyue Li, and Hangyu Mao. Benchmarking the text-to-sql capability of large language models: A comprehensive evaluation. *arXiv preprint arXiv:2403.02951*, 2024.