# OLIVIA: Online Learning via Inference-time Action Adaptation

**Anonymous ACL submission**

## Abstract

Large language models (LLMs) tool-use policies select actions from a finite candidate set at each reasoning step. We observe that this action-selection interface is already a linear contextual scorer: under a restricted output vocabulary, the LM head computes per-action scores from the last-layer hidden state, exactly the structure assumed by linear contextual bandits. Existing online adaptation methods for LLMs operate indirectly through prompts or retrieved memories, leaving this decision layer fixed and providing no explicit mechanism for action-level credit assignment or uncertainty-aware exploration. We propose OLIVIA, a lightweight online adaptation layer that replaces the frozen action-selection head with a per-action contextual linear bandit, selects actions via LinUCB exploration on the LLM's hidden state and updates only its per-action sufficient statistics from step-level feedback via a rank-one Sherman–Morrison rule. Without backbone fine-tuning and additional LLM calls per step, OLIVIA adds negligible overhead relative to LLM inference. Across four tool-use benchmarks TOOLBENCH, TASKBENCH, TASKBENCH-MM, BFCL and two backbones `Qwen3-4B`, `Mistral-7B`, OLIVIA outperforms prompt-based online adaptation and contextual-bandit variants.

## 1 Introduction

Large language models make a sequence of action-selection decisions during tool-use task, at each reasoning step, the model emits a thought and selects an action from a finite candidate set (Yao et al., 2023b; Shinn et al., 2023). In many settings, the same policy repeated queries against a shared tool catalog or batches of similar workflows where feedback from one interaction is informative for the next. Despite this phenomenon, the action selection policy is static at inference time, model follows a fixed prompt-guided distribution, and observed feedback is at most used to modify the textual context rather than the policy itself. A growing line of inference-time adaptation methods (Shinn et al., 2023), CLIN (Majumder et al., 2023), experiential memory (Zhao et al., 2024; Wang et al., 2023), dynamic in-context selection (Wang et al., 2025b) operates indirectly through prompts or memories. These methods change what the model writes, but they expose no explicit decision layer that can score candidate actions, represent uncertainty, or be updated online from action-level feedback. As a result, online improvement remains heuristic and tied to prompt design, not to an optimization mechanism over the policy itself.

We argue that the action-selection layer of an LLM is already a linear contextual scorer. Once the next-token distribution is restricted to a fixed set of candidate actions, the LM head reduces to scoring each action as a linear function of the last-layer hidden state, precisely the structure of a linear contextual bandit (Li et al., 2010; Agrawal and Goyal, 2014). Hence, we propose OLIVIA, a lightweight online adaptation layer that replaces the frozen action-selection head with a per-action contextual linear bandit (Figure 1). OLIVIA uses the LLM's hidden state as bandit context, selects actions via LinUCB-style exploration, and updates its per-action statistics from step-level feedback through a rank-one Sherman–Morrison rule. To stay aligned with the frozen model's prior, the bandit is warm-started from the LLM's own action embeddings. The resulting design adds essentially no cost to the inner loop: no backbone fine-tuning, no gradient backpropagation, and no extra LLM calls per step—each update is a single outer product, far cheaper than a forward pass through the model. The step-level reward that OLIVIA consumes is naturally available in two regimes increasingly common in LLM tool use: *verifier-based reward*, where each candidate action can be pro-

grammatically checked—executable code, API success or failure, unit-test-verifiable outputs, retrieval relevance—as in verifier-driven RL fine-tuning of LLMs (Shao et al., 2024; Wang et al., 2025a); and *labeled bootstrap streams*, where a small set of annotated traces is replayed through the policy to calibrate the action layer before broader use. In both, OLIVIA turns per-step verification into a sample-efficient policy update.

**Contributions.**

- **A bandit view of the LM head.** We show that, once the next-token distribution is restricted to a fixed action set, the frozen LM head is already a linear contextual scorer. It can therefore be swapped for an online-learnable bandit at the same interface, without touching the rest of the model.
- **The OLIVIA layer.** We instantiate this view as a single per-action linear bandit with LinUCB exploration, rank-one Sherman Morrison updates, and warm-starting from the LLM's own action embeddings. The layer adds no gradient updates and no extra LLM calls, and its memory grows linearly in the number of candidate actions.
- **Strong online adaptation at low cost.** On four tool-use benchmarks (TOOLBENCH, TASKBENCH, TASKBENCH-MM, BFCL) and two open-weight backbones (`Qwen3-4B`, `Mistral-7B`), OLIVIA outperforms prompt-based adaptation (CLIN, REFLEXION) and contextual-bandit baselines that use the same hidden-state features, while spending substantially less compute per episode than reflection-style methods.

## 2 Related work

### 2.1 Inference-time adaptation of LLMs.

A growing body of work adapts LLM behavior at inference time without modifying model weights. Most methods operate *indirectly* through prompts, memories, or retrieved context: verbal reflection and episodic memory (Shinn et al., 2023; Majumder et al., 2023; Zhong et al., 2023), retrospective prompt optimization (Yao et al., 2024), dynamic in-context example selection (Wang et al., 2025b), in-context preference adaptation (Wu et al., a), parameter-internalized context (Wang et al., 2025c), reusable natural-language experience or skill libraries (Zhao et al., 2024; Wang et al., 2023; Xia et al., 2025), CoT debiasing and offline alignment (Wu et al., 2024b, 2025), information-theoretic prompt tuning (Wu et al., 2023), and lightweight test-time adaptation under environment shift (Chen et al., 2026; Wu et al., b; Yu et al., 2025). Surveys document the rapid expansion of these methods across personalized, multimodal, and federated LLMs (Zhang et al., 2024; Wu et al., 2024a,c; Surana et al., 2026; Huang et al., 2025). These approaches change the model's textual input, but the action-selection layer itself remains fixed, providing no mechanism for per-action scoring, uncertainty estimation, or principled online updates from action-level feedback. OLIVIA is complementary: it leaves prompts and reasoning generation untouched, and instead replaces the frozen action head with an online learnable bandit layer.

### 2.2 Contextual bandits and linear UCB.

Linear contextual bandits are a classical model for sequential decision making under partial feedback. The LinUCB algorithm (Li et al., 2010) maintains per-action sufficient statistics and selects actions by an upper-confidence rule on a linear reward model. Subsequent work extends linear bandits to nonlinear scoring (Ban et al., 2023), clustered or hierarchical structure (Wu et al., 2021; Hu et al., 2025), conversational and recommendation settings (Xia et al., 2023), and delayed or context-coupled feedback (Zhang et al., 2025; Levy et al., 2025). We observe that the LLM's action-selection head is a per-action linear scorer over the last hidden state is simply to the parametric structure these methods assume. OLIVIA instantiates LinUCB on this interface, warm-started from the LLM's own action embeddings, so the bandit inherits the LLM's prior over actions rather than starting from a random initialization.

### 2.3 Verifier-based reward for LLMs.

A parallel line of work optimizes LLMs with rewards derived from automated verifiers rather than human preferences: rule-based correctness for math and code (Shao et al., 2024; Wang et al., 2025a), RL-style optimization with verifiable outcomes (Schulman et al., 2017; Mundada et al., 2026), and related preference-optimization variants (Rafailov et al., 2023; Li et al., 2025; Kveton et al., 2025). This regime supplies per-step or per-trajectory rewards without requiring human label-

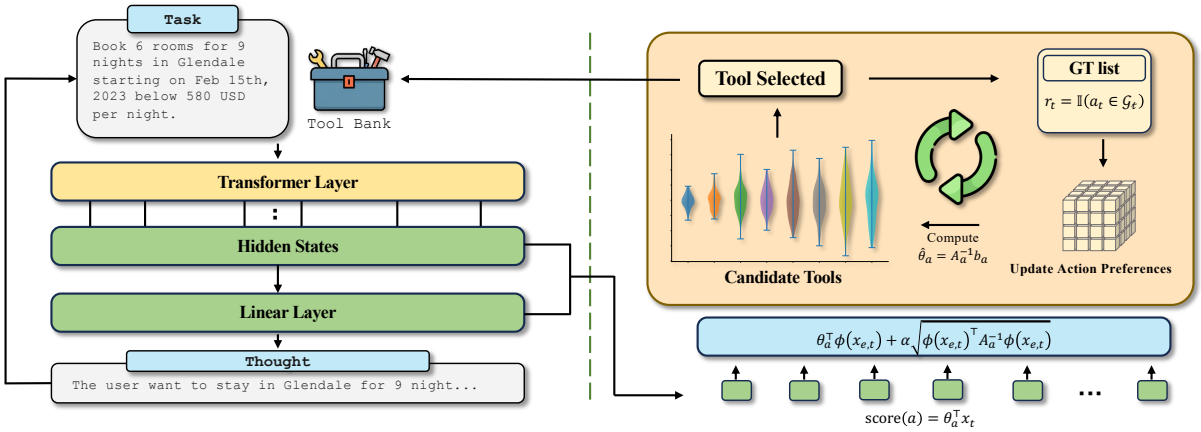


Figure 1: Overview of OLIVIA. **Left**: A frozen LLM backbone processes the task and trajectory prefix through its transformer layers, producing hidden states and a reasoning trace (`Thought`). **Right**: At each action-selection step, OLIVIA extracts the last-layer hidden state as the decision context and scores candidate tools using per-action UCB estimates.

ing, and is the same regime that makes OLIVIA's online updates practical. Whereas these methods use verifier signals as a training-time objective for full backbone fine-tuning, OLIVIA uses them as an inference-time signal for rank-one Sherman Morrison update to a single linear layer, sidestepping gradient computation through the backbone entirely.

### 2.4 Online optimization of sequential LLM decisions.

A complementary line treats inference itself as an online optimization problem over intermediate decisions. Examples include explicit search or planning over thought states (Yao et al., 2023a; Hao et al., 2023; Zhou et al., 2024; Koh et al., 2026), adaptive test-time compute allocation (Inoue et al., 2025; Liu et al., 2024), and uncertainty-aware decoding or sampling (Cai et al., 2025). These approaches optimize over the reasoning trace or compute schedule; OLIVIA optimizes over the action-selection layer with a lightweight bandit on top of a frozen backbone.

## 3 Preliminaries

### 3.1 Contextual Bandits

A contextual bandit models repeated decision making under partial feedback (Li et al., 2010; Ban et al., 2023). At round $n = 1, 2, \ldots, N$, the learner observes a context $x_n \in \mathcal{X}$, chooses an action $a_n \in \mathcal{A}(x_n)$, and receives a scalar reward $r_n$ for that action alone; the rewards of unchosen actions are not observed. This per-round observability of the chosen-action reward is the standard partial-feedback assumption of contextual bandits and is what OLIVIA will exploit at the LLM action-selection interface. An optimal action satisfies

$$a_n^* \in \arg\max_{a \in \mathcal{A}(x_n)} \mathbb{E}[r_n \mid x_n, a].$$

The learner's goal is to find an action-selection rule that achieves high cumulative reward across rounds while balancing *exploration* of uncertain actions against *exploitation* of actions that already appear promising. When the expected reward is modeled as a parametric function of contextual features, the learner can generalize across related decision states and employ principled exploration strategies such as upper confidence bounds (Ban et al., 2023).

### 3.2 ReAct-Style Tool-Use Policies

We consider an LLM tool-use policy that interacts with an environment over multiple decision steps following the ReAct paradigm (Yao et al., 2023b; Zhou et al., 2024; Koh et al., 2026). In episode $e$, the policy receives a task input $q_e$ and a candidate action set $\mathcal{A}_e$ (the available tools). Let $h_{e,t-1}$ denote the interaction history before step $t$, comprising prior reasoning traces, selected actions, and environment observations. At step $t$ the model selects an action $a_{e,t} \in \mathcal{A}_{e,t}$, receives an observation $o_{e,t}$, and updates the history:

$$h_{e,t} = h_{e,t-1} \oplus (a_{e,t}, o_{e,t}),$$

where $\oplus$ denotes concatenation. The episode terminates when the model emits a stop action or reaches a step budget. Because the history grows

with each step, every action decision is conditioned on a distinct, trajectory-dependent context $x_{e,t} = (q_e, h_{e,t-1})$.

## 4 Problem Formulation

We study *online inference-time action adaptation* for LLMs used as ReAct-style tool-use policies: a frozen language model should improve its action choices over a stream of related episodes by updating only a lightweight decision layer from interaction feedback, without fine-tuning the backbone.

### 4.1 ReAct Action Selection as a Contextual Bandit

Consider a stream of ReAct-style episodes $e = 1, 2, \ldots, N$. Using the notation of Section 3.2, at step $t$ of episode $e$ the policy faces a decision context $x_{e,t} = (q_e, h_{e,t-1})$ and must choose an action from $\mathcal{A}_{e,t}^{\text{valid}} \subseteq \mathcal{A}_e$. After acting, it observes feedback only for the selected action—the rewards of alternative actions remain unknown.

This per-step structure directly mirrors a contextual bandit (Section 3.1): a context-dependent choice from a discrete action set under partial feedback. Crucially, although full trajectories differ across tasks, *similar local decision states recur* across related episodes, so feedback from earlier decisions remains informative for later ones. We therefore cast each ReAct action-selection step as one round of an online contextual decision problem.

### 4.2 The LLM Action-Selection Layer as a Linear Bandit

The bandit view above needs both a context representation and a per-action scoring rule; the frozen LLM already supplies both. At the token position immediately preceding the next action, the last-layer hidden state

$$u_{e,t} = h_{\text{LLM}}(q_e, h_{e,t-1}) \in \mathbb{R}^d \tag{1}$$

encodes the task and the trajectory so far. Restricting the output vocabulary to the candidate action set $\mathcal{A}_{e,t}^{\text{valid}}$, the LM head reduces to a per-action linear score

$$s_0(u_{e,t}, a) = w_a^\top u_{e,t}, \tag{2}$$

where $w_a \in \mathbb{R}^d$ is the frozen row of the LM head (or an embedding derived from the same model) associated with action $a$.

The frozen LLM's action-selection mechanism is therefore already a *contextual linear scorer*, exactly the structure assumed by linear contextual bandits. We are not claiming the full LLM is a bandit; we are isolating its *action-selection interface* as a linear layer that can be separated from the underlying reasoning dynamics. The same linear structure also admits a learnable replacement $\theta_a$ that starts from the LLM's prior $w_a$ and is updated online from interaction feedback.

### 4.3 Online Adaptation and Exploration

Replacing the frozen scoring weights with learnable parameters $\{\theta_a\}$ creates an online learning problem under partial feedback: the learner must estimate which actions yield high reward in each context, but observes feedback only for the chosen one. Always picking the currently best-looking action risks never discovering that an under-explored alternative is superior.

Prompt-based adaptation methods such as reflection (Shinn et al., 2023) and memory augmentation (Majumder et al., 2023) influence behavior indirectly through context manipulation and have no mechanism for systematic exploration or for quantifying uncertainty over candidate actions. The linear bandit structure of Section 4.2 resolves this directly: an explicit parametric model yields both an expected reward and an uncertainty estimate for each action, so an optimistic selection rule can add an exploration bonus proportional to that uncertainty. As feedback accumulates the uncertainty shrinks and the learner concentrates on high-reward actions—balancing exploration and exploitation automatically.

## 5 Methodology

This section presents OLIVIA, the concrete instantiation of the framework in Section 4. OLIVIA operates entirely at the action-selection interface of a frozen LLM: a lightweight online decision layer picks the next action from the valid candidate set using the model's hidden state. All backbone parameters, prompts, and reasoning generation remain fixed; only the decision layer is updated from online feedback.

### 5.1 Hidden-State Decision Layer

At decision step $t$ of episode $e$, we extract the LLM's last-layer hidden state at the token position

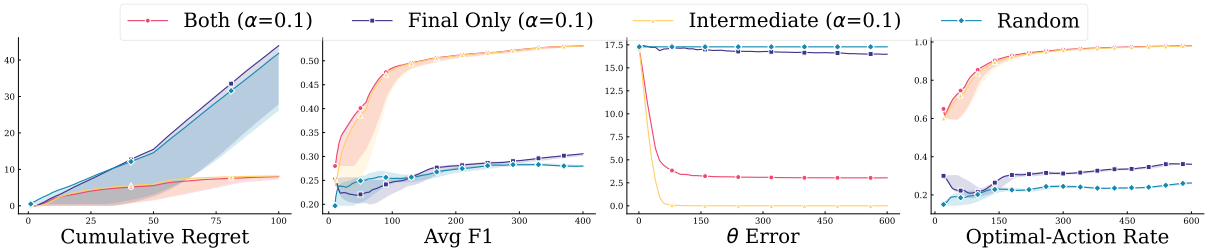


Figure 2: Synthetic validation. Left: cumulative regret over rounds. Right: parameter estimation error $\|\hat{\theta}_t - \theta^\star\|_2^2$. OLIVIA converges faster and recovers the true reward parameter more accurately than weaker online baselines.

immediately preceding the next action token:

$$u_{e,t} = h_{\text{LLM}}(q_e, h_{e,t-1}) \in \mathbb{R}^d. \tag{3}$$

This vector serves as the contextual input to the online controller; it encodes both task semantics and the evolving reasoning history, so similar local decision states yield similar contexts and enable generalization across episodes.

For each candidate action $a$, OLIVIA maintains per-action sufficient statistics

$$A_a \in \mathbb{R}^{d\times d}, \qquad b_a \in \mathbb{R}^d, \qquad \hat{\theta}_a = A_a^{-1} b_a, \tag{4}$$

where $\hat{\theta}_a$ is the current estimate of the action's reward parameter, yielding the base predicted reward $s_{e,t}(a) = \hat{\theta}_a^\top u_{e,t}$.

**LLM-aligned initialization.** Rather than starting from a random policy, we warm-start the controller from the frozen model's own prior:

$$A_a^{(0)} = I, \qquad b_a^{(0)} = \text{embed}(a), \tag{5}$$

where $\text{embed}(a) \in \mathbb{R}^d$ is the mean input embedding of the action's name tokens under the same LLM. With $A_a^{(0)} = I$, the initial parameter satisfies $\hat{\theta}_a^{(0)} = \text{embed}(a)$, so the controller's starting scores are aligned with the frozen model's representation of each action—inheriting the LLM's semantic prior before any feedback has been collected.

### 5.2 UCB-Based Exploration

Because feedback is observed only for the selected action (Section 4.3), we adopt LinUCB (Li et al., 2010), which augments the predicted reward with a context-dependent uncertainty bonus:

$$\text{UCB}_{e,t}(a) = \hat{\theta}_a^\top u_{e,t} + \alpha\sqrt{u_{e,t}^\top A_a^{-1} u_{e,t}}, \tag{6}$$

with exploration coefficient $\alpha > 0$. The first term is the current reward estimate; the second is the width of the confidence interval for action $a$ along the current context. The selected action is

$$a_{e,t} \in \arg\max_{a\in\mathcal{A}_{e,t}^{\text{valid}}} \text{UCB}_{e,t}(a). \tag{7}$$

This bonus has two useful properties: it is *context-dependent*, so the controller explores actions that are uncertain in the current state rather than uniformly at random; and it *shrinks automatically* as each observation tightens the confidence ellipsoid, so well-characterized actions stop being explored. By contrast, $\varepsilon$-greedy ignores the context, Thompson sampling adds stochastic noise, and prompt-based adaptation has no explicit exploration mechanism.

The valid set $\mathcal{A}_{e,t}^{\text{valid}} \subseteq \mathcal{A}_e$ can shrink within an episode: once an action has been selected its maximum allowed number of times, it is removed from the candidate set. This prevents degenerate repetition while still permitting repeated use when warranted by the verifier. Smaller $\alpha$ weights the current estimate and the LLM-aligned prior more heavily; larger $\alpha$ encourages broader early exploration. We study sensitivity to $\alpha$ in Section 6.3.

### 5.3 Step-Level Feedback and Online Update

**Reward signal.** OLIVIA consumes a binary step-level reward $r_{e,t} \in \{0, 1\}$ produced by a per-step *verifier*: an automated correctness check that depends only on the current action and state, not on a gold trajectory. This is the same form of supervision used by recent rule-based RL fine-tuning of LLMs (Shao et al., 2024; Wang et al., 2025a). In our tool-use experiments, the verifier is instantiated from each benchmark's reference tool multiset $G_e$:

$$r_{e,t} = \mathbf{1}[a_{e,t} \in G_{e,t}^{\text{rem}}], \tag{8}$$

where $G_{e,t}^{\text{rem}}$ is the portion of $G_e$ not yet consumed by earlier steps. When $r_{e,t} = 1$, one occurrence of $a_{e,t}$ is removed from $G_{e,t}^{\text{rem}}$, so each reference

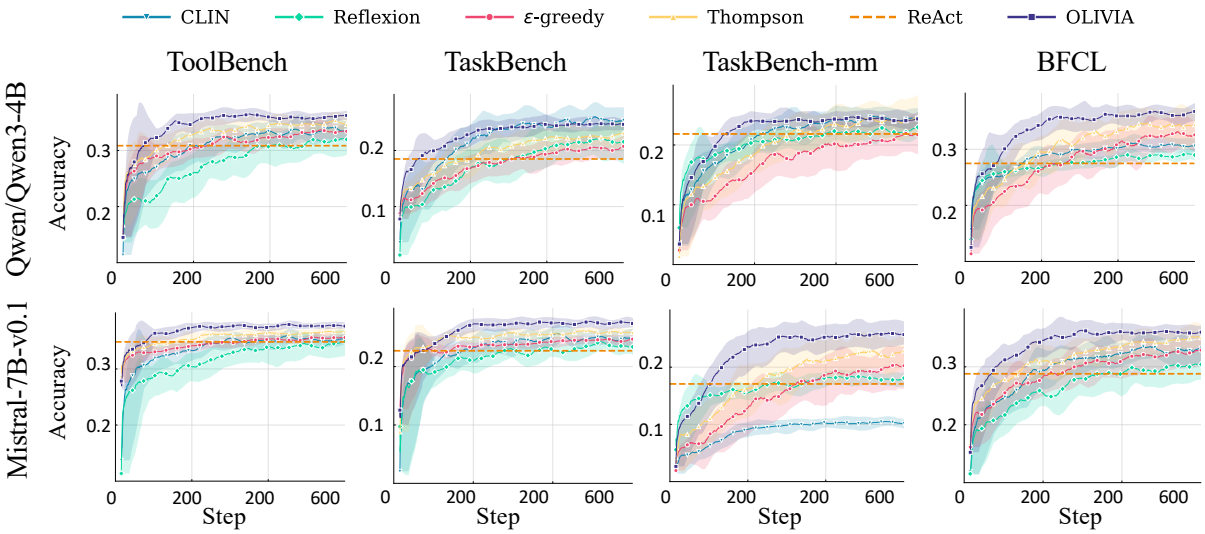


Figure 3: Running-average F1 over the episode stream on the four benchmarks. Static baselines (CoT, BM25, ReAct) remain flat; online methods improve over the stream. OLIVIA converges faster and reaches higher final F1 than CLIN.

tool can contribute to the reward at most once per episode. The update rule is otherwise agnostic to the verifier and applies unchanged to code execution, API success/failure, unit tests, or human step feedback.

**Online update.** After observing $r_{e,t}$, only the statistics of the selected action are updated via a rank-one rule:

$$A_{a_{e,t}} \leftarrow A_{a_{e,t}} + u_{e,t} u_{e,t}^{\top}, \qquad b_{a_{e,t}} \leftarrow b_{a_{e,t}} + r_{e,t}\, u_{e,t}, \tag{9}$$

giving $\hat{\theta}_{a_{e,t}} = A_{a_{e,t}}^{-1} b_{a_{e,t}}$. We maintain $A_a^{-1}$ incrementally via the Sherman–Morrison formula, so each update runs in $O(d^2)$ time without re-inverting the full covariance matrix. An episode terminates on a stop action, an empty valid set, or step-budget exhaustion; the objective across the stream is cumulative reward.

### 5.4 Validation on Synthetic Streams

Before applying OLIVIA to real tool-use streams, we verify that the controller actually recovers the underlying decision structure in a setting where the ground truth is known. We construct synthetic contextual decision streams of 600 rounds with $|\mathcal{A}|$=10 candidate actions and $d$=20-dimensional random context vectors; at each round, the reward of each action is linear in the context under an unknown per-action parameter $\theta_a^\star$ drawn at the start of the stream, and the learner observes only the reward of the selected action—matching the partial-feedback structure of ReAct action selection.

Figure 2 reports cumulative regret and parameter estimation error along the stream. OLIVIA achieves lower regret and tighter parameter recovery than weaker online baselines, confirming that the controller converges to the true decision structure rather than improving by chance. This sanity check underwrites the real-world results that follow in Section 6: gains on tool-use benchmarks are consistent with the controller actually learning the underlying reward structure, not merely benefiting from a better prompt.

## 6 Experiments

### 6.1 Experimental Setup

**Benchmarks.** We evaluate on four tool-use benchmarks: TOOLBENCH (Xu et al., 2023), TASKBENCH, TASKBENCH-MM (Shen et al., 2024), and BFCL (Patil et al., 2023). These benchmarks span complementary tool-use settings, varying in task domain, modality, and candidate-set size. Each instance provides a natural-language task, a candidate tool set, and one or more reference tool-call trajectories. Following the formulation in §5.3, we use each benchmark's reference tool annotations to instantiate the per-step verifier: the reward $r_{e,t}$ is positive when the selected action is a reference tool still required at the current step. This is the same form of programmatic, rule-based supervision used by verifier-driven RL fine-tuning of LLMs (Shao et al., 2024; Wang et al., 2025a), and stands in for any per-step correctness signal

| Model | Method | Dataset | | | |
|---|---|---|---|---|---|
| | | ToolBench | TaskBench | TaskBench-MM | BFCL |
| qwen | CoT | 0.211 | 0.030 | 0.031 | 0.082 |
| | BM25 | 0.353 | 0.116 | 0.127 | 0.181 |
| | ReAct | 0.520 | 0.174 | 0.162 | 0.272 |
| | CLIN | 0.553 | 0.211 | 0.170 | 0.307 |
| | Reflexion | 0.537 | 0.195 | 0.164 | 0.292 |
| | $\varepsilon$-greedy (hidden) | 0.558 | 0.189 | 0.164 | 0.339 |
| | Thompson (hidden) | 0.571 | 0.197 | 0.172 | 0.353 |
| | OLIVIA (Ours) | **0.585** | 0.205 | **0.178** | **0.366** |
| mistral | CoT | 0.142 | 0.036 | 0.064 | 0.107 |
| | BM25 | 0.353 | 0.116 | 0.127 | 0.181 |
| | ReAct | 0.386 | 0.178 | 0.146 | 0.281 |
| | CLIN | 0.394 | 0.193 | 0.127 | 0.324 |
| | Reflexion | 0.388 | 0.187 | 0.170 | 0.307 |
| | $\varepsilon$-greedy (hidden) | 0.397 | 0.191 | 0.186 | 0.328 |
| | Thompson (hidden) | 0.408 | 0.199 | 0.196 | 0.341 |
| | OLIVIA (Ours) | **0.420** | **0.208** | **0.207** | **0.352** |

Table 1: Main results (F1) across four benchmarks and two open-weight backbones. First block: non-learning policies (CoT, BM25, ReAct). Second block: prompt-based online adaptation (CLIN, Reflexion). Third block: bandit-family online learners that share OLIVIA's hidden-state features.

computable from the action and current state—for instance, execution success, API status, or unit-test verification.

**Backbone models.** We use `Qwen/Qwen3-4B` (Yang et al., 2025) and `mistralai/Mistral-7B-v0.1` (Jiang et al., 2023) as open-weight backbones that support ReAct-style tool use while remaining practical for repeated online evaluation. We keep the backbone, system prompt, and few-shot demonstrations fixed across all methods so that performance differences can be attributed to the decision policy alone. **Baselines.** We compare against three groups of baselines that match OLIVIA's setup as closely as possible.

*Non-learning policies.* **CoT** (Wei et al., 2023) uses chain-of-thought prompting for free-form reasoning and tool selection. **BM25** (Robertson and Zaragoza, 2009) ranks candidate tools by lexical similarity to the task description. **ReAct** (Yao et al., 2023b) performs iterative reasoning and tool selection but does not adapt across episodes.

*Prompt-based online adaptation.* **CLIN** (Majumder et al., 2023) maintains a persistent textual memory updated after each episode. **Reflexion** (Shinn et al., 2023) generates verbal self-reflections after failed episodes and prepends them to subsequent prompts.

*Bandit-family variants on the same hidden-state features.* To isolate the contribution of UCB-style exploration, we include two online learners that consume the same last-layer hidden state $u_{e,t}$ as OLIVIA and maintain the same per-action sufficient statistics $(A_a, b_a)$, but differ only in the selection rule. **$\varepsilon$-greedy (hidden)** selects the highest-scoring action with probability $1 - \varepsilon$ and a uniformly random valid action otherwise. **Thompson Sampling (hidden)** draws $\tilde{\theta}_a \sim \mathcal{N}(\hat{\theta}_a, A_a^{-1})$ at each step and selects the action with the highest sampled score.

**Online protocol.** Each dataset is treated as a stream of episodes. For every seed, we randomly shuffle the episode order and run each method sequentially over that stream, reporting mean and standard deviation over three random seeds. All methods use the same per-episode step budget $B = |G| + s$, where $|G|$ is the reference tool-multiset size and $s$ is a small fixed allowance for recovery from early mistakes, and each tool is capped at $m$ selections per episode to prevent degenerate repetition. **Evaluation Metrics.** For each episode we compute precision, recall, and F1 between the policy's tool calls and the benchmark's reference calls, evaluating functional equivalence as in Patil et al. (2023). We report macro-averaged scores over episodes with F1 as the primary metric. To support our efficiency claims, we further report per-method prompt tokens injected by the online-learning state, LLM calls per episode, and wall-clock per episode. Cumulative regret and parameter estimation error

| Method | Growth | 100 | 1k | 10k |
|---|---|---|---|---|
| CLIN | $O$(ep) | 10k | 100k$^\dagger$ | 1M$^\dagger$ |
| Reflexion | $O$(fail ep) | 5k | 50k$^\dagger$ | 500k$^\dagger$ |
| OLIVIA | 0 | 0 | 0 | 0 |

(a) Prompt tokens injected by the online-learning state at 100/1k/10k episodes. $^\dagger$ exceeds a typical 32k context window.

| Method | calls/ep ↓ | s/ep ↓ |
|---|---|---|
| ReAct | 5.0 | $28.0 \pm 3.2$ |
| CLIN | 5.0 | $34.5 \pm 4.1$ |
| Reflexion | 6.4 | $41.5 \pm 5.8$ |
| OLIVIA | 5.0 | $28.2 \pm 3.2$ |

(b) Per-episode compute on BFCL with `Qwen3-4B` (mean $\pm$ std over 100 episodes).

Table 2: Efficiency of online-learning methods: (a) prompt-state growth; (b) per-episode compute.

$\|\hat{\theta}_t - \theta^\star\|_2^2$ are reported in the synthetic validation of Section 5.4.

### 6.2 Main Results

**Performance.** Table 1 reports F1 across all four benchmarks on both backbones. OLIVIA outperforms every baseline group: it improves substantially over the strongest static policy (REACT), beats prompt-based online adaptation (CLIN, REFLEXION), and—most directly—beats the bandit-family variants ($\varepsilon$-greedy and Thompson Sampling) that share its hidden-state features and sufficient statistics but differ only in the exploration rule. This last comparison isolates the contribution of UCB-style exploration: at fixed features and fixed update rule, principled optimism gives consistent gains over random or stochastic exploration.

**Learning curves.** Figure 3 plots running-average F1 over the episode stream. Static baselines (CoT, BM25, ReAct) produce flat reference lines since their policies are fixed. Among the online methods, CLIN and REFLEXION improve gradually through textual memory and reflection updates, but OLIVIA improves faster and converges to a higher F1 across all four benchmarks. The lightweight decision layer turns sparse step-level verifier signals into steadily better action selection, without modifying the backbone model.

**Efficiency.** Table 2 reports the inference-time cost of each online-learning method. OLIVIA's decision-layer state lives entirely off-prompt, while CLIN and REFLEXION accumulate prompt-injected memory that overflows a 32k context within $\sim$1k episodes (Table 2a). On wall-clock (Table 2b), OLIVIA matches REACT to within $\sim$1%, while CLIN and REFLEXION pay growing per-call overhead and additional reflection-generation calls, respectively. The accuracy gains in Table 1 therefore come at essentially ReAct-level inference cost, and at substantially lower cost than reflection-style online adaptation.

Table 3: Reward-design ablation on synthetic streams.

| Reward | Performance | | Convergence | |
|---|---|---|---|---|
| | **F1** ↑ | **Opt.%** ↑ | **Reg@100** ↓ | $\|\theta - \theta^*\|^2$ ↓ |
| Final | 0.323 | 36.2 | 56.3 | 16.48 |
| Step | 0.536 | 98.0 | 8.2 | **0.00** |
| Both | **0.536** | **98.2** | **8.3** | 3.04 |

### 6.3 Ablation Studies

**Effect of $\alpha$.** We sweep $\alpha \in \{0.1, 0.5, 1.0\}$ on the synthetic streams of Section 5.4. The intermediate-only panels of Figure 12 show that OLIVIA is robust across this range: smaller $\alpha$ leans on the LLM-aligned prior, while larger $\alpha$ explores more aggressively at the cost of slower concentration on high-reward actions. We use $\alpha = 0.1$ in the main experiments.

**Effect of reward design.** On the same streams, Table 3 shows that step-level feedback substantially outperforms a final-only reward, confirming the value of local credit assignment in multi-step tool selection. Combining the two signals performs comparably to step-level alone, indicating that the dense step signal already supplies most of the learnable structure.

## 7 Conclusion

We presented OLIVIA, a lightweight online adaptation layer that turns the frozen LM head into a per-action contextual linear bandit on the last-layer hidden state. The construction rests on a simple observation, once the candidate set is restricted, the LM head is already a linear contextual scorer, so the action-selection interface can be replaced by an online-learnable bandit while leaving the backbone untouched, yielding gradient-free updates and no extra LLM calls per step. The results position the learnable action-selection layer as a complement to, rather than a replacement for prompt-based online adaptation for LLMs.

## Limitations

OLIVIA assumes a finite candidate action set and access to the backbone's last-layer hidden states; both are satisfied by open-weight tool-use stacks but not by closed-model APIs. Per-action memory scales as $O(d^2)$, which is parallelizable across actions and lives outside GPU memory, but invites future work on low-rank or shared-feature parameterizations for very large action sets. We also focus on a verifier-based per-step reward; extending OLIVIA to noisier or fully environment-derived feedback signals is a natural next step.

**LLM usage:** AI writing tools were used to assist with drafting and the icons in Figure 1 are generated by AI drawing tool.

## References


Shipra Agrawal and Navin Goyal. 2014. Thompson sampling for contextual bandits with linear payoffs. *Preprint*, arXiv:1209.3352.

Yikun Ban, Yunzhe Qi, and Jingrui He. 2023. Neural contextual bandits for personalized recommendation. *Preprint*, arXiv:2312.14037.

Tiffany Tianhui Cai, Hongseok Namkoong, Daniel Russo, and Kelly W Zhang. 2025. Active exploration via autoregressive generation of missing data. *Preprint*, arXiv:2405.19466.

Arthur Chen, Zuxin Liu, Jianguo Zhang, Akshara Prabhakar, Zhiwei Liu, Shelby Heinecke, Silvio Savarese, Victor Zhong, and Caiming Xiong. 2026. Test-time adaptation for llm agents via environment interaction. *Preprint*, arXiv:2511.04847.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *Preprint*, arXiv:2305.14992.

Songwen Hu, Ryan A. Rossi, Tong Yu, Junda Wu, Handong Zhao, Sungchul Kim, and Shuai Li. 2025. Interactive visualization recommendation with hier-SUCB. In *THE WEB CONFERENCE 2025*.

Chengkai Huang, Junda Wu, Yu Xia, Zixu Yu, Ruhan Wang, Tong Yu, Ruiyi Zhang, Ryan A. Rossi, Branislav Kveton, Dongruo Zhou, Julian McAuley, and Lina Yao. 2025. Towards agentic recommender systems in the era of multimodal large language models. *Preprint*, arXiv:2503.16734.

Yuichi Inoue, Kou Misaki, Yuki Imajuku, So Kuroki, Taishi Nakamura, and Takuya Akiba. 2025. Wider or deeper? scaling llm inference-time compute with adaptive branching tree search. *Preprint*, arXiv:2503.04412.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2026. Tree search for language model agents. *Preprint*, arXiv:2407.01476.

Branislav Kveton, Xintong Li, Julian McAuley, Ryan Rossi, Jingbo Shang, Junda Wu, and Tong Yu. 2025. Active learning for direct preference optimization. *arXiv preprint arXiv:2503.01076*.

Orin Levy, Liad Erez, Alon Cohen, and Yishay Mansour. 2025. Regret bounds for adversarial contextual bandits with general function approximation and delayed feedback. *Preprint*, arXiv:2510.09127.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, page 661–670. ACM.

Xintong Li, Chuhan Wang, Junda Wu, Rohan Surana, Tong Yu, Julian McAuley, and Jingbo Shang. 2025. Importance sampling for multi-negative multimodal direct preference optimization. *arXiv preprint arXiv:2509.25717*.

Jiahao Liu, Qifan Wang, Jingang Wang, and Xunliang Cai. 2024. Speculative decoding via early-exiting for faster llm inference with thompson sampling control mechanism. *Preprint*, arXiv:2406.03853.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2023. Clin: A continually learning language agent for rapid task adaptation and generalization. *Preprint*, arXiv:2310.10134.

Gagan Mundada, Zihan Huang, Rohan Surana, Sheldon Yu, Jennifer Yuntong Zhang, Xintong Li, Tong Yu, Lina Yao, Jingbo Shang, Julian McAuley, and 1 others. 2026. Ws-grpo: Weakly-supervised group-relative policy optimization for rollout-efficient reasoning. *arXiv preprint arXiv:2602.17025*.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *Preprint*, arXiv:2305.15334.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. 2024. Taskbench: Benchmarking large language models for task automation. *Preprint*, arXiv:2311.18760.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Rohan Surana, Gagan Mundada, Xunyi Jiang, Chuhan Wang, Zhenwei Tang, Difan Jiao, Zihan Huang, Yuxin Xiong, Junda Wu, Sheldon Yu, Xintong Li, Raghav Jain, Nikki Kuang, Sizhe Zhou, Bowen Jin, Zhendong Chu, Tong Yu, Ryan Rossi, Kuan-Hao Huang, and 3 others. 2026. Generate, filter, control, replay: A comprehensive survey of rollout strategies for llm reinforcement learning. *Preprint*, arXiv:2605.02913.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *Preprint*, arXiv:2305.16291.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. 2025a. Reinforcement learning for self-improving agent with skill library. *arXiv preprint arXiv:2512.17102*.

Ruoyu Wang, Junda Wu, Yu Xia, Tong Yu, Ryan A Rossi, Julian McAuley, and Lina Yao. 2025b. Dice: Dynamic in-context example selection in llm agents via efficient knowledge transfer. *arXiv preprint arXiv:2507.23554*.

Yu Wang, Xinshuang Liu, Xiusi Chen, Sean OBrien, Junda Wu, and Julian McAuley. 2025c. Self-updatable large language models by integrating context into model parameters. In *International Conference on Learning Representations*, volume 2025, pages 16961–16979.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Junda Wu, Xintong Li, Ruoyu Wang, Yu Xia, Yuxin Xiong, Jianing Wang, Tong Yu, Xiang Chen, Branislav Kveton, Lina Yao, and 1 others. 2025. Ocean: Offline chain-of-thought evaluation and alignment in large language models. In *International Conference on Learning Representations*, volume 2025, pages 100570–100589.

Junda Wu, Hanjia Lyu, Yu Xia, Zhehao Zhang, Joe Barrow, Ishita Kumar, Mehrnoosh Mirtaheri, Hongjie Chen, Ryan A Rossi, Franck Dernoncourt, and 1 others. 2024a. Personalized multimodal large language models: A survey. *arXiv preprint arXiv:2412.02142*.

Junda Wu, Rohan Surana, Zhouhang Xie, Yiran Shen, Yu Xia, Tong Yu, Ryan A Rossi, Prithviraj Ammanabrolu, and Julian McAuley. a. In-context ranking preference optimization. In *Second Conference on Language Modeling*.

Junda Wu, Yuxin Xiong, Xintong Li, Sheldon Yu, Zhengmian Hu, Tong Yu, Rui Wang, Xiang Chen, Jingbo Shang, and Julian McAuley. b. Ctrls: Chain-of-thought reasoning via latent state-transition. In *The 29th International Conference on Artificial Intelligence and Statistics*.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. 2024b. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087.

Junda Wu, Tong Yu, Rui Wang, Zhao Song, Ruiyi Zhang, Handong Zhao, Chaochao Lu, Shuai Li, and Ricardo Henao. 2023. Infoprompt: Information-theoretic soft prompt tuning for natural language understanding. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Junda Wu, Zhehao Zhang, Yu Xia, Xintong Li, Zhaoyang Xia, Aaron Chang, Tong Yu, Sungchul Kim, Ryan A Rossi, Ruiyi Zhang, and 1 others. 2024c. Visual prompting in multimodal large language models: A survey. *arXiv preprint arXiv:2409.15310*.

Junda Wu, Canzhe Zhao, Tong Yu, Jingyang Li, and Shuai Li. 2021. Clustering of conversational bandits for user preference learning and elicitation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2129–2139.

Yu Xia, Yiran Jenny Shen, Junda Wu, Tong Yu, Sungchul Kim, Ryan A Rossi, Lina Yao, and Julian McAuley. 2025. Sand: Boosting llm agents with self-taught action deliberation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3062–3077.

Yu Xia, Junda Wu, Tong Yu, Sungchul Kim, Ryan A Rossi, and Shuai Li. 2023. User-regulation deconfounded conversational recommender system with bandit feedback. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2694–2704.

Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. 2023. On the tool manipulation capability of open-source large language models. *Preprint*, arXiv:2305.16504.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2024. Retroformer: Retrospective large language agents with policy gradient optimization. *Preprint*, arXiv:2308.02151.

Sheldon Yu, Yuxin Xiong, Junda Wu, Xintong Li, Tong Yu, Xiang Chen, Ritwik Sinha, Jingbo Shang, and Julian McAuley. 2025. Explainable chain-of-thought reasoning: An empirical analysis on state-aware reasoning dynamics. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16660–16667, Suzhou, China. Association for Computational Linguistics.

Kelly W. Zhang, Tiffany Tianhui Cai, Hongseok Namkoong, and Daniel Russo. 2025. Contextual thompson sampling via generation of missing data. *Preprint*, arXiv:2502.07064.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, and 1 others. 2024. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. *Preprint*, arXiv:2308.10144.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *Preprint*, arXiv:2305.10250.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language agent tree search unifies reasoning acting and planning in language models. *Preprint*, arXiv:2310.04406.

# A Appendix

This appendix collects material that complements the main paper. Appendix A.1 states the full OLIVIA procedure in pseudocode. Appendix A.2 walks through qualitative TOOLBENCH2 trajectories from early and late in the online stream, showing how the per-action updates reshape tool selection over time. Appendix A.3 reports per-benchmark learning curves for both backbones, and Appendix A.4 reports a reward-design study on the synthetic streams of Section 5.4.

## A.1 Implementation Algorithm

Algorithm 1 states the complete OLIVIA procedure referenced throughout the main paper. Each episode begins by receiving a task and initializing the valid action set. At every decision step, the frozen LLM generates a reasoning trace and the last-layer hidden-state context (line 6); the controller scores all valid actions with LinUCB (line 8) and selects the maximizer (line 10). After observing the binary step-level reward, the selected action's sufficient statistics are updated in place via the rank-one Sherman–Morrison formula (lines 16–17), with no gradient computation or backward pass. The per-action inverse covariance $A_a^{-1}$ and reward accumulator $b_a$ persist across episodes, so the controller continues to improve over the entire online stream.

## A.2 Case Study

We present three qualitative trajectories from the TOOLBENCH2 benchmark (`Qwen3-4B`, $\alpha$=0.1) that illustrate how OLIVIA's online controller reshapes tool selection over the episode stream.

**Same task, before and after learning.** Examples 1 and 2 show the *same task* encountered at episode 3 (early in the stream) and episode 935 (after substantial online updates). At episode 3 the controller has not yet accumulated feedback and selects entirely unrelated tools (F1 = 0.00). By episode 935, per-action updates have raised the UCB scores of the correct tools so that all three ground-truth tools are recovered (recall = 1.00), even though the first three actions are still exploratory misses. The traces visualize per-action credit assignment in action: every miss (✗) updates only the selected tool's statistics, while every hit (✓) raises that tool's UCB score, so subsequent episodes increasingly favor the admissible tools. This per-action adaptation is the mechanism that prompt- and memory-based baselines do not expose, and that the bandit-family variants share with OLIVIA (differing only in the selection rule).

**Example 1 — Episode 3 (early stream): F1**

**Task:** I'm organizing a team building activity for my company. Can you help me create a short URL for the event details? Also, suggest some domain options for the team building link. Furthermore, provide me with a list of available domains for branding the short link.
**Ground truth (3):** bitly.shorten, URL Link Shortener.Create a new link, URL Link Shortener.Get a list of domains

1. *Thought:* The user needs a short URL for the event details.
   **Action:** OTT details.New Arrivals ✗
2. *Thought:* The user also wants domain options for the short link.
   **Action:** Streaming Availability.Get Basic ✗
3. *Thought:* Additionally, the user needs a list of available domains for branding.
   **Action:** Similar Movies.find similar ✗
4. *Thought:* Now, I need to check if all the actions have been completed.
   **Action:** Watchmode.Streaming Release Dates ✗
5. *Thought:* All required actions have been completed.
   **Action:** Vimeo.GetRelatedChannels ✗
6. *Thought:* The user's task is complete.
   **Action:** Movie, TV, music search and download.Get Monthly Top 100 Music Torrents ✗

**Result:** P=0.00, R=0.00, F1=0.00

**Example 2 — Episode 935 (late stream, same task): F1**

**Task:** I'm organizing a team building activity for my company. Can you help me create a short URL for the event details? Also, suggest some domain options for the team building link. Furthermore, provide me with a list of available domains for branding the short link.
**Ground truth (3):** bitly.shorten, URL Link Shortener.Create a new link, URL Link Shortener.Get a list of domains

1. *Thought:* The user needs a short URL for the event details.
   **Action:** Watchmode.Title Seasons ✗
2. *Thought:* The user also wants domain options for the short link.
   **Action:** Watchmode.New Titles ✗
3. *Thought:* Additionally, the user is looking for a list of available domains for branding.
   **Action:** Streaming Availability.Search Pro ✗
4. *Thought:* Now, I need to check if all the actions have been completed.
   **Action:** URL Link Shortener.Create a new link ✓
5. *Thought:* All tasks are completed.
   **Action:** bitly.shorten ✓
6. *Thought:* The final answer is ready.
   **Action:** URL Link Shortener.Get a list of domains ✓

**Result:** P=0.50, R=1.00, F1=0.67

**Generalization to a harder task.** Example 3 shows the controller after further training (episode 1130) on a more complex, previously unseen task whose ground truth contains nine tools. The controller recovers seven of the nine within twelve steps (recall = 0.78), demonstrating that the per-action statistics learned from the easier ToolBench2 stream transfer to harder task phrasings.

**Example 3 — Episode 1130 (complex task, 7 of 9 ground-truth tools recovered): F1**

**Task:** I'm organizing a movie marathon for my company's team-building event. Could you suggest some action movies from the 90s that are available for streaming on Netflix, Prime Video, and Hulu? Please include the IMDb ratings and streaming links for each movie.
**Ground truth (9):** Watchmode.List Titles, Watchmode.New People, Watchmode.Title Seasons, OTT details.Basic Info, OTT details.Advanced Search, OTT details.Title Details, Streaming Availability.Search By Title, Streaming Availability.Search Ultra, Streaming Availability.Search Pro

1. *Thought:* The user is looking for action movies from the 90s.
   **Action:** Vimeo.SearchVideos ✗
2. *Thought:* The user wants to know where these movies are available for streaming.
   **Action:** Watchmode.List Titles ✓
3. *Thought:* The user is interested in IMDb ratings for these movies.
   **Action:** Streaming Availability.Search Ultra ✓
4. *Thought:* The user wants the actual streaming links for each movie.
   **Action:** OTT details.Basic Info ✓
5. *Thought:* The information should be accurate and up-to-date for a company event.
   **Action:** Streaming Availability.Search Pro ✓
6. *Thought:* The information needs to be concise and easy to share with the team.
   **Action:** Watchmode.Title Seasons ✓
7. *Thought:* The user is likely looking for a variety of options to cater to different tastes.
   **Action:** OTT details.Title Details ✓
8. *Thought:* The user might also want to know the release years to ensure they are from the 90s.
   **Action:** OTT details.Search ✗
9. *Thought:* The information should be organized in a way that is easy to use during the marathon.
   **Action:** OTT details.Additional Title Details ✗
10. *Thought:* The user is looking for movies that can be watched in a short time frame.
    **Action:** Streaming Availability.Search By Title ✓
11. *Thought:* The movies should be engaging and suitable for a group setting.
    **Action:** Similar Movies.find similar ✗
12. *Thought:* The user is looking for movies that are well-received with good action elements.
    **Action:** Vimeo.GetChannelVideos ✗

**Result:** P=0.58, R=0.78, F1=0.67

### A.3 Per-Benchmark Learning Curves

We report per-benchmark learning curves for both backbones. Each plot shows running-average F1 over the episode stream: non-learning baselines (CoT, BM25, ReAct) appear as flat reference lines,

while the online methods (CLIN, OLIVIA) evolve as they accumulate feedback. Across all four benchmarks and both backbones, OLIVIA starts near the REACT prior and surpasses CLIN within a few hundred episodes, consistent with the aggregate F1 in Table 1. To keep the figures readable we omit the remaining online baselines from Table 1 (REFLEXION, $\varepsilon$-greedy (hidden), Thompson (hidden)) from the per-benchmark plots; their aggregate scores are reported in Table 1.

**Qwen3-4B.**

**Mistral-7B.**

### A.4 Reward-Design Study

We complement the reward-design ablation in Section 6.3 (Table 3) with stream-level learning curves on the synthetic streams of Section 5.4. Figure 12 reports running-average precision, recall, and F1 under three reward configurations: *step-level only* (the dense per-step verifier reward used throughout the paper), *final only* (a single per-episode binary signal), and *both* (the two combined). Step-level feedback drives substantially faster convergence on all three metrics than the final-only reward, which spreads a single uninformative signal across every selection in the trajectory. Combining the two performs comparably to step-level alone, mirroring the aggregate findings in Table 3.

**Algorithm 1** OLIVIA: Online Contextual Bandit for ReAct Tool Selection

**Inputs**: exploration parameter $\alpha > 0$, extra steps $s$, max selections per arm $m$
**Initialize**: For each arm $a \in \mathcal{A}$: $A_a^{-1} \leftarrow I_d$, $b_a \leftarrow \mathrm{emb}_{\mathrm{LLM}}(a)$, $n_a \leftarrow 0$
**for** episode $e = 1$ to $T$ **do**
  Receive task $q_e$ with ground-truth tool multiset $G_e$; set $\mathcal{V} \leftarrow \mathcal{A}$, $L_e \leftarrow |G_e| + s$
  **for** $t = 1$ to $L_e$ **do**
    Generate thought $z_{e,t}$ via LLM conditioned on $(q_e,\, z_{e,<t},\, a_{e,<t})$
    Extract context $x_{e,t} \leftarrow h_{\mathrm{LLM}}\big(q_e,\, z_{e,1:t},\, a_{e,<t}\big)$ {last hidden state at action position}
    **for all** $a \in \mathcal{V}$ **do**
      $\theta_a \leftarrow A_a^{-1} b_a, \quad \mathrm{ucb}_{e,t}(a) \leftarrow \theta_a^\top x_{e,t} + \alpha \sqrt{x_{e,t}^\top A_a^{-1} x_{e,t}}$
    **end for**
    Select $a_{e,t} \leftarrow \arg\max_{a \in \mathcal{V}} \mathrm{ucb}_{e,t}(a)$
    Execute $a_{e,t}$; observe per-step reward $r_{e,t} = \mathbf{1}\big[a_{e,t} \in G_e^{\mathrm{rem}}\big]$
    **if** $r_{e,t} = 1$ **then**
      Decrement remaining count $G_e^{\mathrm{rem}}(a_{e,t})$
    **end if**
    Update selection count $n_{a_{e,t}} \mathrel{+}= 1$; if $n_{a_{e,t}} \geq m$ remove $a_{e,t}$ from $\mathcal{V}$
    **Policy update** (Sherman–Morrison):
      $k \leftarrow A_{a_{e,t}}^{-1} x_{e,t}$
      $A_{a_{e,t}}^{-1} \leftarrow A_{a_{e,t}}^{-1} - \dfrac{k\,k^\top}{1 + x_{e,t}^\top k}$, $b_{a_{e,t}} \leftarrow b_{a_{e,t}} + r_{e,t}\, x_{e,t}$
    **if** $\mathcal{V} = \emptyset$ **then**
      **break** {all arms exhausted}
    **end if**
  **end for**
  Compute episode metrics: $\mathrm{recall}_e$, $\mathrm{precision}_e$, $\mathrm{F1}_e$
**end for**

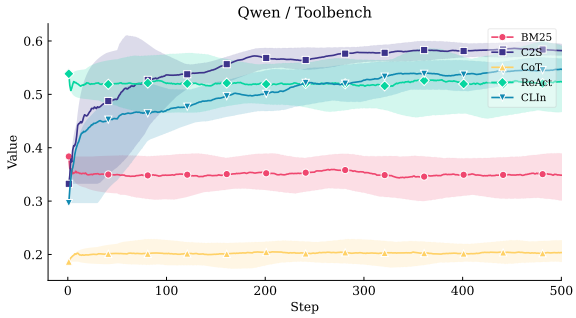


Figure 4: Qwen3-4B on ToolBench.

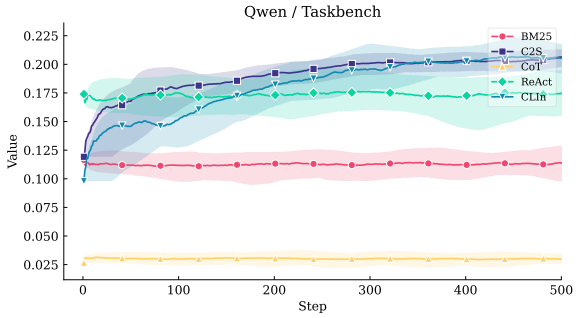


Figure 5: Qwen3-4B on TaskBench.

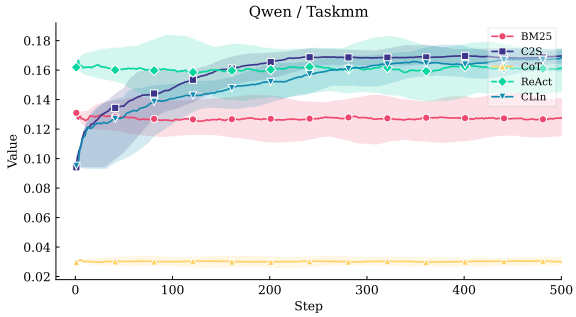


Figure 6: Qwen3-4B on TaskBench-MM.

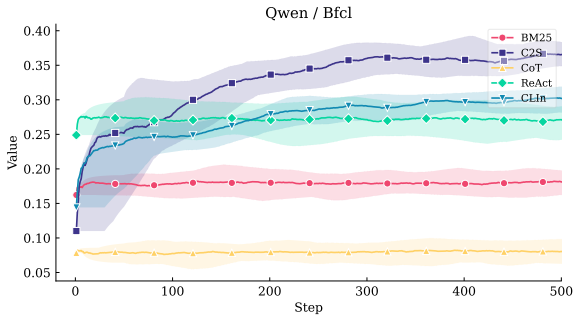


Figure 7: Qwen3-4B on BFCL.

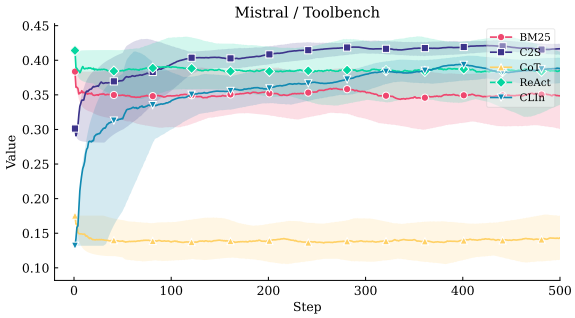


Figure 8: Mistral-7B on ToolBench.

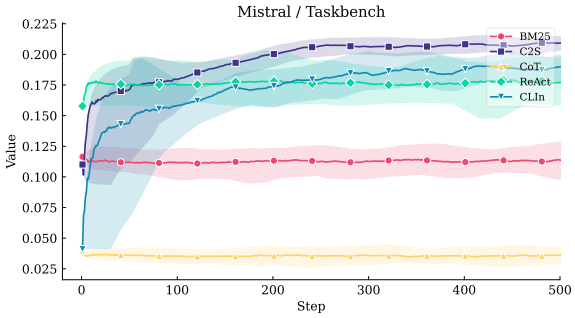


Figure 9: Mistral-7B on TaskBench.

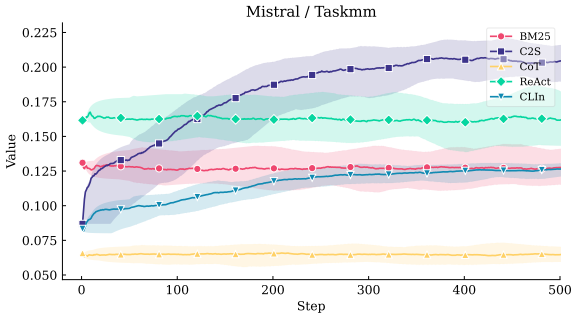


Figure 10: Mistral-7B on TaskBench-MM.

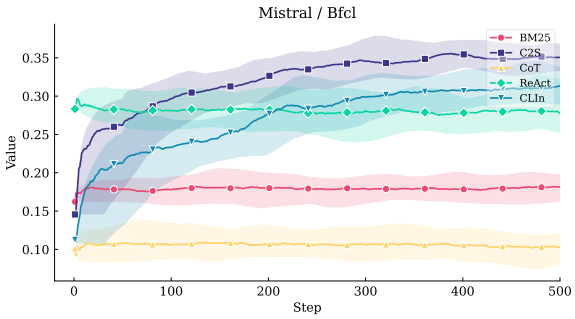


Figure 11: Mistral-7B on BFCL.

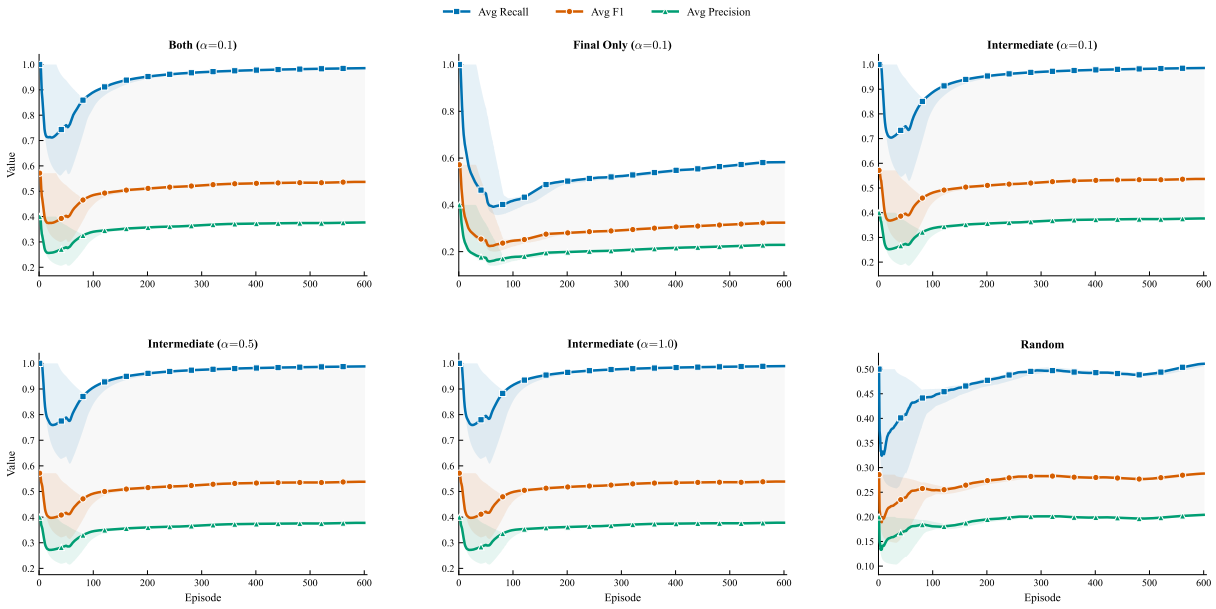


Figure 12: Reward-design study on synthetic streams. Running-average precision, recall, and F1 (left to right) over the episode stream for three reward configurations: *step-level only*, *final only*, and *both*. Shaded bands denote seed-level variability.