

TopoCurate: Modeling Interaction Topology for Tool-Use Agent Training

Jinluan Yang^{1*}, Yuxin Liu^{2*}, Zhengyu Chen^{2*}, Chengcheng Han^{2*},
Yueqing Sun², Qi Gu², Hui Su², Xunliang Cai², Fei Wu¹, Kun Kuang¹

¹Zhejiang University ²Meituan LongCat
yangjinluan@zju.edu.cn

Abstract

Training tool-use agents typically relies on outcome-based filtering: Supervised Fine-Tuning (SFT) on successful trajectories and Reinforcement Learning (RL) on pass-rate-selected tasks. However, this paradigm ignores interaction dynamics—successful trajectories may lack error recovery or exhibit redundancy, while pass rates fail to distinguish structurally informative tasks from trivial ones. To bridge this gap, we propose **TopoCurate**, an interaction-aware framework that shifts the paradigm from linear outcome filtering to topological interaction modeling. By projecting multi-trial rollouts into a unified semantic quotient topology within the same task, we construct a structured manifold that explicitly captures the divergence between success and failure pathways across tool invocations and environmental responses. Leveraging this representation, we introduce a dual-selection mechanism: for SFT, we prioritize trajectories demonstrating reflective recovery, semantic efficiency, and strategic diversity to mitigate covariate shift and mode collapse; for RL, we select tasks with high error branch ratios and strategic heterogeneity, maximizing gradient Signal-to-Noise Ratio to address vanishing signals in sparse-reward settings. Evaluations on BF-CLv3 and Tau2 Bench show that TopoCurate achieves consistent gains of 4.2% (SFT) and 6.9% (RL) over state-of-the-art baselines.

Keywords: Tool-use Agent, Data Curation, Topological Modeling

1 Introduction

The advent of agentic intelligence has shifted the paradigm for large language models (LLMs) from static text generation to proactive engagement in multi-turn, tool-based interactions with environments (Team et al., 2025a; Liu et al., 2025; Yang

et al., 2025a; Team et al., 2026). Consequently, the efficacy of agent training—whether via Supervised Fine-Tuning (SFT) or Reinforcement Learning (RL)—is strictly bottlenecked by the structural quality of the interaction data (Team et al., 2025b). However, prevailing paradigms (Li et al., 2025b; Xu et al., 2025b; Yang et al., 2025b) remain predominantly linear outcome-centric: SFT trajectories are selected based on binary success, and RL tasks by pass-rate thresholds from isolated trajectories. While scalable, these metrics are inherently myopic to interaction dynamics (Gooding and Grefenstette, 2025), treating reasoning processes as flat, *linear chains* and obscuring the critical *structured dependencies*—such as error recovery and state transitions—that govern robust agency.

This reliance on outcomes creates a critical misalignment between data selection and learning value, a phenomenon we term the **Outcome Equivalence Illusion**. Our topological analysis reveals two core deficiencies: (i) *SFT Selection Bias*: Standard outcome filtering selects successful trajectories indiscriminately, ignoring process robustness, efficiency, and strategic diversity. This dilutes critical error-recovery patterns with simple paths (Lan et al., 2025), while redundant loops and homogenized solutions lower training value (Zhu et al., 2025). Consequently, the agent suffers from covariate shift, semantic inefficiency, and mode collapse (Yan et al., 2026), overfitting to narrow behaviors. (ii) *RL Gradient Vanishing*: In sparse-reward RL, tasks with identical pass rates mask divergent structural training potentials. Tasks that yield homogeneous rollouts (identical outcomes regardless of action variations) produce vanishing policy gradients (low Advantage). In contrast, tasks with structural bifurcation—where sharp decision boundaries lead to outcome divergence—provide high-contrast signals, yielding a high Signal-to-Noise Ratio (SNR) for efficient optimization.

The root cause of this illusion is the prevailing

*These authors contributed equally. Corresponding authors: Qi Gu and Kun Kuang.

linear, isolated view of trajectories. Evaluating each rollout independently obscures the latent structure—the branching points, recovery paths, and state convergence—that defines robust agent behavior. To break this illusion, we propose **TopoCurate**, a framework that shifts the paradigm from linear outcome filtering to *topological interaction modeling*. Our core innovation is to transcend the isolated analysis of trajectories by projecting multi-trial rollouts from the same task into a unified semantic topology. By merging semantically equivalent action-observation states across trials, we convert a sparse forest of linear traces into a structured, global manifold—a Directed Acyclic Graph (DAG). This topology explicitly visualizes the divergence between success and failure pathways and densifies visitation statistics, enabling the first rigorous, quantitative analysis of interaction process quality.

Leveraging this topological representation, we introduce a principled, dual-selection mechanism: For **SFT**, we define topological metrics for *Reflective Recovery* (prioritizing error correction), *Semantic Efficiency* (penalizing redundancy), and *Distributional Diversity* (combating mode collapse) to curate a dataset that approximates a robust, strategic expert policy. For **RL**, we propose metrics for *Error Branch Ratio* and *Strategic Heterogeneity* to select tasks that maximize gradient SNR by prioritizing those with high structural complexity and diverse valid solutions, directly addressing the vanishing signal problem.

Our contributions are threefold:

- We introduce **TopoCurate**, a framework that shifts agent data curation from outcome-based filtering to topological interaction modeling. By projecting linear rollouts into a semantic quotient topology, we provide a rigorous formalism for capturing the interactions that outcome metrics neglect.
- We devise a topology-driven dual-selection mechanism theoretically grounded in learning dynamics. For SFT, we introduce *Reflective Recovery*, *Semantic Efficiency*, and *Distributional Diversity* to mitigate covariate shift and mode collapse. For RL, we propose structural task selection based on *Error Branch Ratios* and *Strategic Heterogeneity*, which maximize gradient SNR by prioritizing tasks with critical decision boundaries.
- Extensive evaluations on BFCLv3 and Tau2

bench consistently demonstrate the effectiveness of our proposed method, achieving average gains of 4.2% (SFT) and 6.9% (RL) over state-of-the-art baselines.

2 Related Work

2.1 Tool Use Data Synthesis and Selection

Multi-Agent Simulation (MAS) is the prevailing paradigm for tool-use synthesis (Zhang et al., 2025a), encompassing approaches from social simulation (Tang et al., 2025) to committee-based validation (Prabhakar et al., 2025). While systems like Kimi K2 (Team et al., 2025a) and Simia (Li et al., 2025b) scale MAS or simulate environmental feedback, their data selection remains heuristic and outcome-centric. Pipelines such as APIGen-MT (Prabhakar et al., 2025) and Simia (Li et al., 2025b) rely on terminal checks or rule-based filtering, treating trajectories as flat sequences (Xu et al., 2025a); similarly, MUA (Zhao et al., 2025) selects RL tasks solely by pass-rate. Although works like ToolPRM (Lin et al., 2025) and ToolPRMBench (Li et al., 2026) employ fine-grained process supervision, they focus on optimizing search or reward models rather than fundamentally assessing training data quality. Consequently, current paradigms neglect the interaction topology—its structural decision points and error recoveries—failing to separate robust process quality from stochastic success (Zhang et al., 2026). In contrast, our TopoCurate framework addresses this gap by shifting to process-aware topological modeling, explicitly selecting data that maximizes structural learning value for both SFT and RL without external reward models.

2.2 Tool-Use Evaluation

The evaluation of tool-use agents has progressed from static syntax checks to dynamic, multi-turn interaction. Benchmarks like BFCL (Yan et al., 2024) and ACEBench (Chen et al., 2025a) assess multi-step tool invocation and error correction. TauBench (Yao et al., 2024) and its successor τ_2 -bench (Barres et al., 2025) simulate dual-control environments where user and agent actions co-evolve the state, testing policy adherence in complex domains. Concurrently, evaluation metrics have shifted from greedy accuracy (Pass@1) to distributional robustness (Pass@k) (Yu, 2025; Li et al., 2025b). Pass@k measures an agent’s capacity to discover diverse valid solutions within a stochastic search space, which serves in our work as a proxy for strategic

heterogeneity—indicating whether the model has learned multiple pathways rather than overfitting to a single behavior.

3 Preliminaries

In this section, we formalize the multi-turn tool-use process as a sequential decision-making problem and clarify the data selection of SFT and RL training as our optimization object.

3.1 Definition of Multi-Turn Agentic Tool Use

Interaction Process. We formulate the interaction between a tool-use agent and an environment as a Partially Observable Markov Decision Process (Kurniawati, 2022). Define task as a tuple $\mathcal{T} = (q, \mathcal{C}, \mathcal{E})$, where q denotes the user intent, $\mathcal{C}$ represents the context, and $\mathcal{E}$ is the evaluation protocol. At each step t , the agent receives an observation o_t and generates an action unit $z_t = (r_t, a_t)$, comprising internal reasoning r_t (e.g., Chain-of-Thought) and a structured tool action a_t . The interaction history is denoted as $h_t = (\mathcal{T}, z_1, o_1, \dots, z_{t-1}, o_{t-1})$. The goal is to learn a policy $\pi_\theta(z_t|h_t)$ that maximizes the task success.

Interaction Turns and Semantic Topology. To model the agent-environment feedback loop, we define a complete interaction turn as $\hat{z}_t = (r_t, a_t, o_t)$, comprising the agent’s reasoning, action, and the environmental observation. To capture the latent structure of tasks, we define a semantic equivalence relation over the space of complete turns $\hat{\mathcal{Z}}$. Two turns $\hat{z}_i, \hat{z}_j$ are considered equivalent if their actions and observations are semantically similar, indicating they represent the same logical state. This relation induces a quotient structure, compressing scattered linear trajectories into a unified Directed Acyclic Graph (DAG) where nodes represent clusters of semantically equivalent behaviors.

3.2 Data Selection for Agentic Tool Use

We consider a scenario with a large-scale task pool $\mathcal{Q}_{\text{pool}}$ and a set of synthesized trajectories. Our goal is to select high-quality subsets to optimize the agent’s performance.

Trajectory Selection for SFT. Let $\mathcal{D}_{\text{pool}} = \{(\mathcal{T}_i, \tau_{i,j})\}$ be a dataset containing multiple candidate trajectories for each task. The data selection problem seeks a subset $\mathcal{D}_{\text{select}} \subset \mathcal{D}_{\text{pool}}$ (represented by weights $w_{i,j} \in [0, 1]$) to maximize validation performance under a budget constraint B :

$$\begin{aligned} \max_w \quad & \mathbb{E}_{(\mathcal{T}, \tau) \in \mathcal{D}_{\text{val}}} [R(\tau \mid \pi_{\theta^*})] \\ \text{s.t.} \quad & \theta^* = \arg \min_{\theta} \sum_{i,j} w_{i,j} \mathcal{L}_{\text{SFT}}(\pi_{\theta}, \tau_{i,j}), \\ & \sum_{i,j} w_{i,j} \leq B. \end{aligned} \quad (1)$$

Limitations. Conventional filtering sets $w_{i,j} = \mathbb{I}[R(\tau_{i,j}) = 1]$. However, this linear outcome approach ignores the interaction topology that may lead to model collapse and covariate shift, e.g., A successful trajectory may still contain redundant loops, lack error recovery mechanisms, or represent a narrow solution path. The detailed discussions can be shown in Appendix C.2. Our objective is to derive a topological prior $w_{i,j}$ that harmonizes execution optimality with process robustness, semantic efficiency, and strategic diversity.

Task Selection for RL. This objective is to curate a subset $\mathcal{Q}_{\text{RL}} \subset \mathcal{Q}_{\text{pool}}$ that maximizes gradient efficiency. Adopting Group Relative Policy Optimization (GRPO) as the training algorithm, the policy is updated by sampling a group of K trajectories for each task and maximizing the relative advantage as Eq. 2, where $\hat{A}(\tau_k)$ is the standardized advantage from sparse outcome rewards.

$$\begin{aligned} J(\theta) = \mathbb{E}_{\mathcal{T} \sim P_{\text{select}}(\mathcal{T})} \mathbb{E}_{\{\tau_{1:K}\} \sim \pi_{\theta}(\cdot|\mathcal{T})} \left[\frac{1}{K} \sum_{k=1}^K \frac{\pi_{\theta}(\tau_k)}{\pi_{\text{old}}(\tau_k)} \hat{A}(\tau_k) \right] \end{aligned} \quad (2)$$

Limitations. Standard approaches define $P_{\text{select}}(\mathcal{T})$ based solely on pass rates. However, this overlooks structural training potential. Tasks yielding homogeneous rollouts produce vanishing advantages ($\hat{A} \approx 0$), providing minimal gradient information (low Signal-to-Noise Ratio, SNR). The detailed discussions can be shown in Appendix C.3. Our framework constructs $P_{\text{select}}(\mathcal{T})$ to prioritize tasks with high Error Branch Ratios and Strategic Heterogeneity, maximizing the contrastive signal of the policy gradient within the topology.

4 Methodology

4.1 Overview

As illustrated in Figure 1, our proposed framework operates in three systematic stages: (i) **Topological Modeling (Sec. 4.2)**: We aggregate raw interaction turns into a Directed Acyclic Graph (DAG),

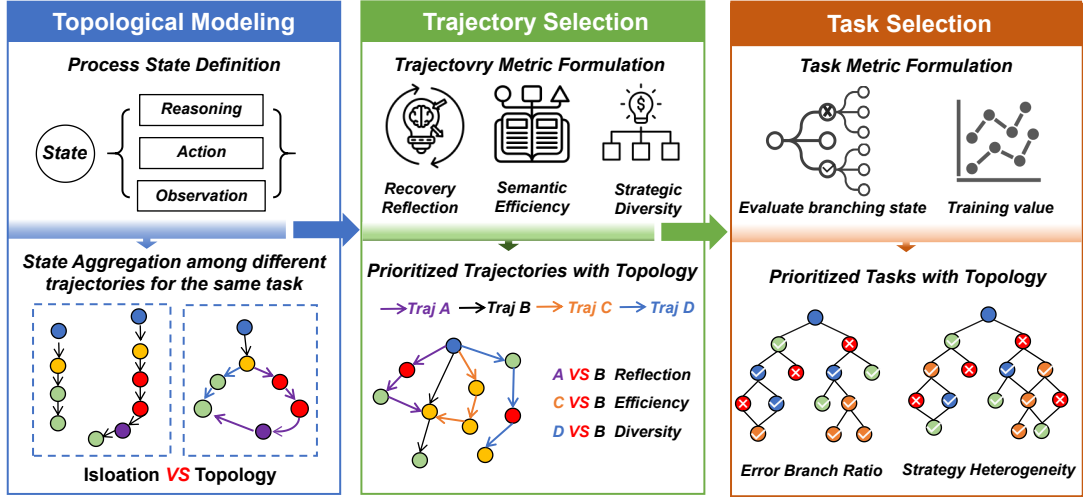


Figure 1: **Overview of the TopoCurate Framework.** Our method operates in three stages: **(Left) Topological Modeling** transforms disjoint rollouts into a unified state-transition graph by defining states via action-observation tuples and aggregating semantically equivalent turns; **(Middle) Trajectory Selection for SFT** applies three process-aware metrics—Reflective Recovery (resilience), Semantic Efficiency (economy), and Strategic Diversity (exploration)—to prioritize high-quality trajectories; **(Right) Task Selection for RL** evaluates tasks using Error Branch Ratio and Strategy Heterogeneity to select structurally complex tasks that maximize gradient efficiency.

defining states by both tool actions and environmental observations. This structure reveals the *Success Potential* of each state; **(ii) Trajectory Selection for SFT (Sec. 4.3):** Instead of simple outcome filtering, we select trajectories based on three topological process metrics—*Reflective Recovery* (resilience), *Semantic Efficiency* (economy), and *Strategic Diversity* (exploration)—to distill a robust expert policy; **(iii) Task Selection for RL (Sec. 4.4):** We prioritize tasks that exhibit high structural complexity. By filtering for high *Error Branch Ratios* and *Strategic Heterogeneity*, we select tasks that provide sharp decision boundaries and diverse solution paths, maximizing training efficiency in sparse-reward settings.

4.2 Modeling Agentic Dynamics via Quotient Topology

Unlike static reasoning chains, agentic tool use is driven by the environment’s feedback loop. While the agent outputs $z_t = (r_t, a_t)$, the semantic state is determined jointly by the action and the environment’s response o_t . To rigorously evaluate this process, we define a *complete interaction turn* as $\hat{z}_t = (r_t, a_t, o_t)$ and compress the space of these turns $\hat{\mathcal{Z}}$ into a dense state space $\mathcal{V}$ via a quotient map $\pi : \hat{\mathcal{Z}} \rightarrow \mathcal{V}$.

Why Merge? The Necessity of State Aggregation. In long-horizon agentic tasks, the action space is combinatorial, and slight variations in tool

arguments result in a disjoint, sparse forest. This sparsity prevents statistical analysis. By defining a semantic equivalence relation $\sim$ and merging equivalent turns, we achieve two critical goals:

- *Densification for Potential Estimation:* Aggregating sparse visits into dense node clusters allows us to robustly estimate the Success Potential Field $\Phi(v) = P(\text{Success}|v)$. Without merging, $\Phi(v)$ would be binary or undefined.
- *Revealing Causal Convergence:* Merging reveals states reached by diverse paths. This “equifinality” uncovers the underlying causal structure of the environment, enabling us to compare the efficiency of different strategies (e.g., shortcuts vs. detours) that arrive at the same intermediate state.

Observation-Driven State Abstraction. A key insight is that agent state is critically defined by observation o_t . For instance, in retail scenarios, `modify_order(id="123")` leads to distinct states depending on whether o_t returns “Success” or “Error: User Confirmation Required”. Let $\text{sim}(\cdot, \cdot)$ denote the cosine similarity between embeddings. We define the equivalence relation $\hat{z}_i \sim \hat{z}_j$ as a conjunction of view-specific constraints:

$$\hat{z}_i \sim \hat{z}_j \iff (\text{sim}_{\text{tool}}(a_i, a_j) \geq \delta_{\text{tool}}) \wedge (\text{sim}_{\text{result}}(o_i, o_j) \geq \delta_{\text{result}}). \quad (3)$$

We utilize a high-performance embedding model (e.g., jina-embeddings-v2-base-en (Günther et al., 2023) and Qwen3-0.6B-Embedding (Zhang et al., 2025b)) with default thresholds ($\delta_{\text{tool}} \approx 0.95, \delta_{\text{result}} \approx 0.90$) to ensure the reasonability of quotient topology. More detailed robustness analysis can be shown in Appendix B.3.

4.3 Topological Trajectory Selection for SFT

Standard SFT filtering selects all trajectories with $R(\tau) = 1$, implicitly assuming a uniform quality distribution. To solve the constrained optimization problem defined in Eq. (2), we propose a second-stage refinement that approximates the optimal weights $w(\tau)$ using topological signals.

Metric Formulation. We derive three topological signals to quantify process quality:

- *Reflective Recovery* (S_{ref}): Prioritizes trajectories where the agent recovers from negative feedback (Dip-Recovery Pattern). A reflection event occurs when potential drops ($\Phi(v_t) \ll \Phi(v_{t-1}) - \epsilon_{\text{dip}}$) and is subsequently restored:

$$S_{\text{ref}}(\tau) = \sum_{t \in \text{jumps}} \frac{\Phi(v_{t+k}) - \Phi(v_t)}{k}. \quad (4)$$

This metric captures agent resilience in dual-control dynamics. For instance, in a telecom task, a trajectory encountering a Destination Unreachable error but recovering via device reboot yields high S_{ref} , whereas abandoning the task incurs penalty.

- *Semantic Efficiency* (S_{eff}): Penalizes redundant loops by comparing actual path length to the geodesic distance on the graph:

$$S_{\text{eff}}(\tau) = \min_{(u,v) \in \text{Convergence}(\tau)} \frac{\mathcal{D}_{\mathcal{G}}(u,v)}{\text{len}(\tau_{u \rightarrow v})}. \quad (5)$$

This serves as a regularizer. In an airline booking context, it penalizes agents that redundantly query `get_flight_details` for the same ID without state changes, favoring concise information gathering.

- *Distributional Diversity* (S_{rare}): Maximizes entropy by weighting the inverse branch popularity $P(v)$ with the local success rate $\text{Acc}(v)$:

$$S_{\text{rare}}(\tau) = \frac{1}{|\tau|} \sum_{v \in \tau} \frac{\text{Acc}(v)}{\log(1 + P(v))}. \quad (6)$$

This metric prevents mode collapse by prioritizing trajectories that traverse rare but successful decision branches, ensuring the policy learns diverse strategies.

Trajectory Selection Strategy. To balance these objectives, we define the selection weight $w(\tau)$ as:

$$w(\tau) = \lambda_{\text{ref}} \cdot \tilde{S}_{\text{ref}}(\tau) + \lambda_{\text{rare}} \cdot \tilde{S}_{\text{rare}}(\tau) + \lambda_{\text{eff}} \cdot \tilde{S}_{\text{eff}}(\tau), \quad (7)$$

where $\tilde{S}$ denotes the normalized score (via z-score standardization across the candidate pool), and $\{\lambda_{\text{ref}}, \lambda_{\text{rare}}, \lambda_{\text{eff}}\}$ are hyperparameters that control the relative importance of each metric. We select trajectories based on their composite scores $w(\tau)$, ensuring a balanced representation of robustness, diversity, and efficiency. Detailed hyperparameter sensitivity analysis are provided in Appendix B.4.

4.4 Topological Task Selection for RL

Standard task selection based solely on pass rates overlooks structural complexity. We propose a topology-driven task selection mechanism.

Metric Formulation. We evaluate the structural complexity of each task’s solution space:

- *Error Branch Ratio* (V_{struct}): The proportion of decision branches leading to failure (where potential drops below a threshold ϵ_{fail}), representing contrastive difficulty:

$$V_{\text{struct}}(\mathcal{T}) = \frac{1}{|\mathcal{B}|} \sum_{v \in \mathcal{B}} \frac{|\{u \in \text{children}(v) : \Phi(u) < \epsilon_{\text{fail}}\}|}{|\text{children}(v)|}. \quad (8)$$

High V_{struct} indicates critical decision nodes. In a retail “Modify Order” task, the state bifurcates at the `check_status` node: respecting the “Shipped” status leads to success, while ignoring it triggers an irreversible failure branch. Such sharp contrasts provide high-fidelity gradient signals for policy adherence.

- *Strategic Heterogeneity* (V_{div}): The Unique Chain Ratio (UCR), measuring the number of valid distinct tool sequences. High UCR indicates multiple valid workflows, preventing mode collapse:

$$V_{\text{div}}(\mathcal{T}) = \frac{|\{\tau : R(\tau) = 1\}|}{|\{\tau : \text{sampled}\}|}. \quad (9)$$

For example, a flight rebooking request might be solvable via two distinct topological paths: a direct `modify_flight` call or a composite `cancel` followed by `book`. Training on such tasks fosters policy plasticity and generalization.

Table 1: **Baseline Comparison.** All methods start with outcome-based filtering; TopoCurate further refines via topological metrics.

Method	Trajectory Selection (SFT)	Task Selection (RL)
APIGEN-MT (Prabhakar et al., 2025)	Outcome ($r = 1$)	N/A
MUA (Zhao et al., 2025)	Outcome ($r = 1$)	Pass Rate
Simia-Tau (Li et al., 2025b)	Outcome ($r = 1$)	N/A
TopoCurate (Ours)	Outcome ($r = 1$) ↓ Topological $S_{ref}, S_{eff}, S_{rare}$	Pass Rate ↓ Topological V_{struct}, V_{div}

Task Selection Strategy. We construct the task sampling distribution $P_{\text{select}}(\mathcal{T})$ to favor structural richness:

$$P_{\text{select}}(\mathcal{T}) \propto \exp\left(\frac{V_{\text{struct}}(\mathcal{T}) + \alpha V_{\text{div}}(\mathcal{T})}{T}\right), \quad (10)$$

where α is a balancing coefficient and T is the temperature.

5 Experiments

5.1 Experimental Setup

Backbones. We employ the Qwen3 series (Yang et al., 2025a) (thinking mode, scaling from 8B to 32B) as our foundational backbones.

Seed Data Construction. We construct the initial pool using a pipeline adapted from APIGen-MT (Prabhakar et al., 2025), tailored to Tau2 Environment. Tasks are stratified by Qwen3-32B’s pass rate: fundamental tasks (high pass rate) are distilled into expert trajectories via Claude-4.5-Sonnet (Anthropic, 2024) to seed the SFT pool, while complex tasks (moderate pass rate) are reserved for the RL pool. Crucially, this generation establishes a raw candidate set. TopoCurate is subsequently applied to select SFT trajectory and RL tasks. The Details are provided in Appendix B.1.

Baselines. We select baselines related to τ -bench environment: (i) **APIGEN-MT** (Prabhakar et al., 2025), which employs LLM committee validation for successful trajectories; (ii) **MUA** (Zhao et al., 2025), which selects trajectories via outcome filtering; and (iii) **Simia-Tau** (Li et al., 2025b), which applies rule-based post-validation for correctness. While all methods perform outcome-based filtering as the primary selection criterion, TopoCurate adopt a **two-stage refinement** (Table 1). After the initial outcome filter following previous works, we construct a quotient topology from the candidate set and apply process-aware metrics ($S_{ref}, S_{eff}, S_{rare}$

for SFT; V_{struct}, V_{div} for RL) to select data with high structural quality. This design rigorously isolates the contribution of topological curation and verifies its flexibility.

Evaluation Benchmarks. We evaluate our approach on two challenging benchmarks: (i) **Tau2 Bench (In-Domain)**. For each test, we conduct 8 repeated tests to achieve pass@1, pass@4, pass@8 for stable evaluation; and (ii) **BFCL v3 Multi-Turn (Out-of-Domain)**, the Berkeley Function Calling Leaderboard, in order to keep the same settings with baselines (Zhao et al., 2025) to assess the model’s generalization across unseen APIs and long interactions. Moreover, we report the results on the advanced **BFCL v4** in Appendix B.6 to further verify the effectiveness of our method.

5.2 Main Results

TopoCurate can consistently achieve better performance and generalization than previous methods due to the topological refinement. As shown in Table 2, our empirical evaluation confirms that TopoCurate consistently establishes new state-of-the-art performance, outperforming leading baselines on both in-domain and OOD benchmarks. Specifically, (i) *Superiority over In-Domain Baselines*: Despite sharing the same τ -bench generation environment as APIGen-MT and Simia-Tau, TopoCurate-SFT achieves higher Pass@1 scores across all domains. This validates our core premise that *data quality, defined by topological richness, outweighs mere generation quantity*. By filtering out redundant or spurious trajectories, TopoCurate distills a denser training signal, enabling robust learning from fewer examples; and (ii) *Generalization to OOD Tasks*: On the BFCL v3 benchmark, our method demonstrates superior generalization (e.g., lower parameter miss rates), indicating that the reasoning primitives learned from topologically curated data are transferable to unseen APIs, rather than being mere domain heuristics.

Pass@k Analysis and Scaling Behavior. We evaluate Pass@k performance across sampling budgets (Figure 2) to validate TopoCurate’s robustness. Key findings include: (i) *Superior Intrinsic Capability*. TopoCurate-SFT achieves substantially higher Pass@1 baselines across all domains. For 32B models, gains over the base model reach +23% (Airline: 0.53 vs. 0.43), +17% (Retail: 0.62 vs. 0.53), and +116% (Telecom: 0.54 vs. 0.25). Similar improvements hold for 8B models (+43% Air-

Table 2: Performance comparison on Tau2 Bench (IID) and BFCL v3 Multiturn (OOD). Models are trained on Tau2 environments. Gray rows denote TopoCurate variants. **Bold** indicates best results.

Model	Tau2 Bench (IID)			BFCL V3 Multiturn (OOD)			
	Airline	Retail	Telecom	Base	Long Context	Miss Func	Miss Param
32B Size Models							
Qwen3-32B-Thinking	0.435	0.525	0.248	0.340	0.210	0.155	0.245
APIGen-MT-SFT	0.347	0.592	0.202	0.375	0.215	0.175	0.205
MUA-SFT	0.311	0.582	0.193	0.350	0.195	0.210	0.285
MUA-RL	0.454	0.673	0.283	0.420	0.215	0.200	0.300
TopoCurate-SFT (w/o Topology)	0.483	0.591	0.521	0.410	0.235	0.125	0.245
TopoCurate-SFT	0.526	0.623	0.539	0.430	0.265	0.135	0.295
TopoCurate-RL (w/o Topology)	0.531	0.643	0.721	0.450	0.245	0.165	0.260
TopoCurate-RL	0.558	0.687	0.784	0.480	0.295	0.225	0.325
8B Size Models							
Qwen3-8B-Thinking	0.297	0.389	0.244	0.230	0.135	0.045	0.135
APIGen-MT-SFT	0.229	0.491	0.126	0.245	0.140	0.050	0.170
MUA-SFT	0.160	0.314	0.090	0.240	0.100	0.110	0.165
MUA-RL	0.190	0.498	0.218	0.210	0.110	0.115	0.150
Simia-Tau-SFT	0.395	0.522	0.145	0.045	0.020	0.005	0.080
Simia-Tau-RL	0.405	0.529	0.156	0.040	0.025	0.015	0.075
TopoCurate-SFT (w/o Topology)	0.436	0.500	0.418	0.265	0.135	0.075	0.150
TopoCurate-SFT	0.445	0.529	0.435	0.265	0.125	0.090	0.160
TopoCurate-RL (w/o Topology)	0.467	0.518	0.533	0.255	0.145	0.105	0.160
TopoCurate-RL	0.481	0.545	0.595	0.295	0.160	0.135	0.180

line, +33% Retail, +83% Telecom), demonstrating that topological curation fundamentally enhances model capability rather than merely improving sampling efficiency. (ii) Consistent Gains at Higher Budgets. As sampling increases (Pass@4, Pass@8), TopoCurate maintains its advantage. In Telecom (32B), TopoCurate’s Pass@8 (0.89) outperforms w/o Topology (0.88) and substantially exceeds the base model (0.51), indicating enhanced distribution quality over viable solution paths. These results validate that topological curation distills robust expert policies capable of adaptive navigation across diverse solution trajectories. Further evidence from Appendix 5 corroborates these findings.

5.3 Ablation Studies

Trajectory Selection for SFT: We dissect the contribution of our three topological SFT metrics (Table 3). Removing *Reflective Recovery* consistently degrades performance, validating that “dip-recovery” patterns are essential for resilience. *Efficiency* plays a critical role in Retail, where its removal drops accuracy by 6.1% (0.529 → 0.468), while *Diversity* is pivotal for Airline, with a 5.2% drop (0.445 → 0.393). The full TopoCurate method harmonizes these objectives, achieving the highest average accuracy (0.470), confirming that an optimal policy must balance resilience, diversity, and

economy; **Task Selection for RL:** As shown in Table 4, we observe that filtering by *Structural Complexity* (V_{struct}) is the dominant factor. Tasks with high error branch ratios provide the necessary contrastive signals—distinguishing optimal actions from critical failure modes—that are absent in trivial tasks. *Strategic Heterogeneity* (V_{div}) acts as a vital regularizer, preventing the policy from overfitting to a narrow set of solution paths.

5.4 Model Behavior and Training Dynamics

Model Behavior Analysis. We analyze three behavioral dimensions as Figure 3 in Appendix. (i) *Reflection*: TopoCurate-SFT agents exhibit markedly higher reflection scores, particularly in Telecom where the reflection rate increases from ~ 0.15 to ~ 0.28 , confirming the internalization of self-correction mechanisms; (ii) *Efficiency*: We observe a shift from fragile brevity to robust persistence. Measured by turns per success (lower is better), our method reduces interaction turns across domains by roughly ~ 0.5 – 2 turns per success, indicating higher efficiency at a superior success rate; (iii) *Strategic Plasticity*: The consistently higher Unique Chain Ratio indicates that our topological diversity metric effectively mitigates mode collapse, fostering a policy capable of adaptively selecting from a diverse repertoire of tool chains.

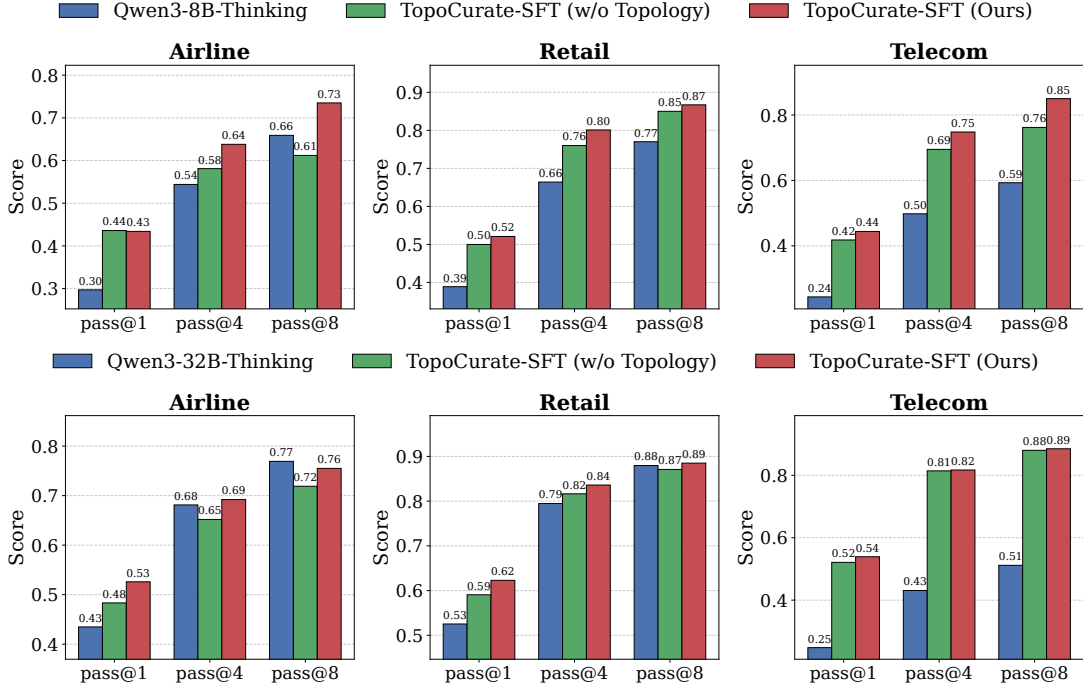


Figure 2: **Pass@k Comparison across three settings:** the Qwen3-Instruct backbone (Thinking mode), a baseline SFT model trained on standard outcome-filtered data (w/o Topology), and our TopoCurate-SFT. The results demonstrate that TopoCurate consistently achieve higher pass@k across all domains and model scales.

RL Training Dynamics Analysis. We conduct two controlled experiments to validate topological curation’s impact on SFT initialization and RL task selection from the training dynamic perspective. (i) *SFT Initialization.* We compare RL runs from different SFT checkpoints (TopoCurate-selected vs. outcome-filtered) using the same RL task pool. Figure 5 (left) shows the topologically-initialized model achieves a higher starting reward (0.29 vs. 0.22) and superior results (0.63 vs. 0.61 at step 60). Evaluation accuracy (Figure 4, left) confirms consistent gains across domains; (ii) *RL Task Selection.* Starting from the same SFT checkpoint, we compare topologically-selected tasks versus uniform sampling. Figure 5 (right) reveals identical starting points (~ 0.25) but divergent outcomes (0.65 vs. 0.61 at step 60). Evaluation accuracy (Figure 4, right) shows w/ Topology consistently outperforms across domains (Airline: 0.545 vs. 0.535; Retail: 0.642 vs. 0.632; Telecom: 0.565 vs. 0.551 at step 60). These experiments across two stages consistently verify the effectiveness of TopoCurate.

5.5 Complexity and Robustness Analysis.

We provide complexity analysis for introduced topology modeling in Appendix C.1. We can observe that the extra 10%-30% data selection time is clearly less than the basic outcome filtering time.

Table 3: **Ablation Study (SFT).** Impact of topological metrics on Pass@1 accuracy (Qwen3-8B-Thinking as backbone). “Ref”, “Div”, and “Eff” denote Reflection, Diversity, and Efficiency.

Method	Components			Tau2 Pass@1			
	Ref	Div	Eff	Airline	Retail	Telecom	Avg
TopoCurate-SFT (w/o Topology)	×	×	×	0.436	0.500	0.418	0.451
w/o Reflection	×	✓	✓	0.436	0.510	0.422	0.456
w/o Diversity	✓	×	✓	0.393	0.472	0.428	0.431
w/o Efficiency	✓	✓	×	0.401	0.468	0.435	0.435
TopoCurate-SFT (Ours)	✓	✓	✓	0.445	0.529	0.435	0.470

Table 4: **Ablation Study (RL).** Impact of topological metrics on Pass@1 accuracy (Qwen3-32B-Thinking as backbone). “Struct” and “Div” denote Error Branch Ratio and Strategic Heterogeneity.

Method	Components		Tau2 Pass@1			
	Struct	Div	Airline	Retail	Telecom	Avg
TopoCurate-RL (w/o Topology)	×	×	0.531	0.643	0.721	0.632
w/o Structure	×	✓	0.542	0.661	0.748	0.650
w/o Diversity	✓	×	0.547	0.668	0.756	0.657
TopoCurate-RL (Ours)	✓	✓	0.558	0.687	0.784	0.676

More importantly, the rl task set could also be reduced to 75% compared with outcome filtering, which will reduce RL training time by 30% while achieving superior training performance. Moreover, the selection of embedding models and metric weights for data curation are also discussed in Appendix to strengthen the robustness of our method.

6 Conclusion

In this work, we introduce TopoCurate, a data curation framework shifting the paradigm from outcome filtering to topological interaction modeling. By projecting linear rollouts into a quotient topology, TopoCurate breaks the outcome equivalence illusion to distill robust behaviors—such as reflective recovery—ignored by standard metrics. Our dual-stage curation mitigates covariate shift in SFT training and maximizes gradient SNR in RL training, establishing new state-of-the-art performance on Tau2 and BFCLv3 benchmarks. This work establishes structure-aware curation as a key driver in building more resilient and adaptable agents.

Limitations

We conduct experiments in tool-use settings, including multi-turn function calling and API interaction, which demonstrate the effectiveness of TopoCurate in structured agent-environment scenarios. Nevertheless, its applicability to other agentic settings, such as search, remains to be further explored. Since these scenarios often involve longer contexts, extending embedding-based topology modeling to them may require more careful investigation. We leave this direction to future work.

Acknowledgements

This work was supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), Research Project of the Philosophy and Social Sciences Planning Research Platform (Laboratory) of Zhejiang Province, and Key R&D Program of Hangzhou (2025SZDA0254).

References

- Anthropic. 2024. Claude 4 model family. <https://www.anthropic.com>. Claude Sonnet 4.5.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2 -bench: Evaluating conversational agents in a dual-control environment. *arXiv:2506.07982*.
- Shihao Cai, Runnan Fang, Jialong Wu, Baixuan Li, Xinyu Wang, Yong Jiang, Liangcai Su, Liwen Zhang, Wenbiao Yin, Zhen Zhang, et al. 2025. Autoforge: Automated environment synthesis for agentic reinforcement learning. *arXiv:2512.22857*.
- Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, et al. 2025a. Acebench: Who wins the match point in tool usage? *arXiv:2501.12851*.
- Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. 2025b. Revisiting scaling laws for language models: The role of data quality and training strategies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23881–23899.
- Mahir Labib Dihan, Tanzima Hashem, Mohammed Eunus Ali, and Md Rizwan Parvez. 2025. Weboperator: Action-aware tree search for autonomous agents in web environment. *arXiv:2512.12692*.
- Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, et al. 2025. Towards general agentic intelligence via environment scaling. *arXiv:2509.13311*.
- Sian Gooding and Edward Grefenstette. 2025. Interaction dynamics as a reward signal for llms. *arXiv:2511.08394*.
- Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, et al. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv:2310.19923*.
- Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. 2025. Tree search for llm agent reinforcement learning. *arXiv:2509.21240*.
- Hanna Kurniawati. 2022. Partially observable markov decision processes and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):253–277.
- Li-Cheng Lan, Andrew Bai, Minhao Cheng, Cho-Jui Hsieh, and Tianyi Zhou. 2025. Exploring expert failures improves llm agent tuning. *arXiv:2504.13145*.
- Dawei Li, Yuguang Yao, Zhen Tan, Huan Liu, and Ruocheng Guo. 2026. Toolprmbench: Evaluating and advancing process reward models for tool-using agents. *arXiv:2601.12294*.
- Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, et al. 2025a. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv:2508.17445*.
- Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Pooven-dran, Robert Sim, and Saravan Rajmohan. 2025b. Simulating environments with reasoning models for agent training. *arXiv:2511.01824*.

- Jianghao Lin, Yuanyuan Shi, Xin Peng, Renjie Ding, Hairui Wang, Yuxuan Peng, Bizhe Bai, Weixi Song, Fengshuo Bai, Huacan Chai, et al. 2025. Toolprm: Fine-grained inference scaling of structured outputs for function calling. *arXiv:2510.14703*.
- Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv:2512.02556*.
- Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, et al. 2025. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv:2504.03601*.
- Leyang Shen, Yang Zhang, Chun Kai Ling, Xiaoyan Zhao, and Tat-Seng Chua. 2025. Carl: Critical action focused reinforcement learning for multi-step agent. *arXiv:2512.04949*.
- Michael Sullivan, Mareike Hartmann, and Alexander Koller. 2025. Procedural environment generation for tool-use agents. *arXiv:2506.11045*.
- Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Tian Jin, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. 2025. Synthesizing post-training data for llms through multi-agent simulation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23306–23335.
- Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. 2025a. Kimi k2: Open agentic intelligence. *arXiv:2507.20534*.
- Meituan LongCat Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chen Gao, Chen Zhang, Chengcheng Han, et al. 2026. Longcat-flash-thinking-2601 technical report. *arXiv:2601.16725*.
- MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, et al. 2025b. Miro-thinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv:2511.11793*.
- Hieu Tran, Zonghai Yao, and Hong Yu. 2025. Exploiting tree structure for credit assignment in rl training of llms. *arXiv:2509.18314*.
- Siqi Wang, Zhengyu Chen, Teng Xiao, Zheqi Lv, Jinluan Yang, Xunliang Cai, Jingang Wang, and Xiaomeng Li. 2025. Scaling and transferability of annealing strategies in large language model training. *arXiv:2512.13705*.
- Feijie Wu, Weiwu Zhu, Yuxiang Zhang, Soumya Chatterjee, Jiarong Zhu, Fan Mo, Rodin Luo, and Jing Gao. 2025. Portool: Tool-use llm training with rewarded tree. *arXiv:2510.26020*.
- Shangyu Xing, Siyuan Wang, Chenyuan Yang, Xinyu Dai, and Xiang Ren. 2025. Lookahead tree-based rollouts for enhanced trajectory-level exploration in reinforcement learning with verifiable rewards. *arXiv:2510.24302*.
- Zengzhuang Xu, Bingguang Hao, Zechuan Wang, Yuntao Wen, Xinyi Xu, Yang Liu, Long Chen, Dong Wang, Maolin Wang, Tong Zhao, et al. 2025a. Funreason-mt technical report: Advanced data synthesis solution for real-world multi-turn tool-use. *arXiv:2510.24645*.
- Zhangchen Xu, Adriana Meza Soria, Shawn Tan, Anurag Roy, Ashish Sunil Agrawal, Radha Pooven-dran, and Rameswar Panda. 2025b. Toucan: Synthesizing 1.5 m tool-agentic data from real-world mcp environments. *arXiv:2510.01179*.
- Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Berkeley function calling leaderboard. https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html. Accessed: 2024-01-16.
- Minghao Yan, Bo Peng, Benjamin Coleman, Ziqi Chen, Zhouhang Xie, Zhankui He, Noveen Sachdeva, Isabella Ye, Weili Wang, Chi Wang, et al. 2026. Pacevolve: Enabling long-horizon progress-aware consistent evolution. *arXiv:2601.10657*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv:2505.09388*.
- Chen Yang, Ran Le, Yun Xing, Zhenwei An, Zongchao Chen, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2025b. Toolmind technical report: A large-scale, reasoning-enhanced tool-use dataset. *arXiv:2511.15718*.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv:2406.12045*.
- Yang Yu. 2025. Pass@ k metric for rlvr: A diagnostic tool of exploration, but not an objective. *arXiv:2511.16231*.
- Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. 2025a. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv:2505.00024*.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang,

Dayiheng Liu, Junyang Lin, et al. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv:2506.05176*.

Zhiwei Zhang, Fei Zhao, Rui Wang, Zezhong Wang, Bin Liang, Jiakang Wang, Yao Hu, Shaosheng Cao, and Kam-Fai Wong. 2026. Robust tool use via fission-grpo: Learning to recover from execution errors. *arXiv:2601.15625*.

Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. 2025. Mua-rl: Multi-turn user-interacting agent reinforcement learning for agentic tool use. *arXiv:2508.18669*.

Wenlin Zhong, Jinluan Yang, Yiquan Wu, Yi Liu, Jianhang Yao, and Kun Kuang. 2026. Sight: Reinforcement learning with self-evidence and information-gain diverse branching for search agent. *arXiv:2602.11551*.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, et al. 2025. Where llm agents fail and how they can learn from failures. *arXiv:2509.25370*.

A More Details about Related Work

A.1 Discussion with Structured Rollout Strategy

Regarding Structured Rollout Strategies, current approaches have developed a highly efficient “Online Exploration Engine.” Techniques such as tree-based sampling and prefix sharing ((Ji et al., 2025; Li et al., 2025a; Tran et al., 2025; Wu et al., 2025)), along with lookahead branching and critical node search ((Xing et al., 2025; Dihan et al., 2025; Shen et al., 2025; Zhong et al., 2026)), have significantly broadened the scope of exploration and made preliminary strides in addressing the challenge of granular credit assignment in sparse reward settings. However, these methods are fundamentally centered on Online Local Optimization, typically treating tree structures as transient vehicles for immediate gradient estimation or trajectory generation. This focus leaves a critical gap in the Offline Global Valuation of interaction data. As a result, valuable computational resources are often wasted on semantically redundant branches or samples with low signal-to-noise ratios.

In contrast, our TopoCurate framework introduces a novel approach with Offline Topological Analysis. By aggregating and projecting historical interaction trajectories into a unified Semantic Quotient Topology, we can precisely pinpoint high-value structures characterized by reflective recovery from a global perspective in an offline setting, thereby drastically improving data utilization efficiency. Consequently, our method enables a profound synergy with existing online strategies. The frontend leverages Structured Rollout techniques to efficiently generate rich, tree-based candidate trajectories, while the backend employs TopoCurate for offline structural distillation. This “Online Generation + Offline Selection” paradigm ensures that the model updates its parameters solely based on the most topologically instructive paths, thereby maximizing the marginal benefit of training.

A.2 Discussion with Environment Scaling

Current approaches to environment scaling, such as AutoForge (Cai et al., 2025), RandomWorld (Sullivan et al., 2025), and AgentScaler (Fang et al., 2025), have successfully established a powerful expansion engine. By leveraging automated environment synthesis, procedural task generation, and API community detection techniques, these methods have tackled the challenges of limited inter-

action scenarios and insufficient domain coverage in agent training. However, they are still confined by an outcome-centric filtering paradigm, where SFT trajectories are selected based solely on binary success signals, and RL tasks are chosen based on pass-rate thresholds.

In contrast, our TopoCurate framework fundamentally disrupts this outcome equivalence illusion by introducing topological interaction modeling. This innovation shifts the focus of selection from “whether it succeeds” to “how it succeeds.” Specifically, TopoCurate projects multi-trial trajectories into a unified Semantic Quotient Topology, constructing Directed Acyclic Graphs (DAGs) to rigorously capture decision bifurcations, error recovery paths, and the causal structure of state transitions—dynamics that existing outcome-based metrics often overlook.

As a result, our method perfectly complements environment scaling pipelines. While expansion techniques provide a vast and varied pool of tasks, TopoCurate performs a structural distillation, ensuring that synthetic data is not ingested blindly. Instead, it is carefully filtered through behavioral dynamics, effectively transforming the benefits of environment scaling into robust decision-making capabilities for agents.

B More Details about Experiments

B.1 Seed Data Construction and Topological Filtering

This appendix provides implementation details for the seed data construction pipeline.

Multi-Agent Synthesis Framework : We adapt a multi-agent collaborative framework inspired by APIGen-MT (Prabhakar et al., 2025) to generate task blueprints and trajectories. The system employs seven specialized agents coordinated by a MasterAgent as follows:

- **UserProfileAgent**: Generates diverse user personas with domain-specific consumption histories.
- **InstructionAgent**: Creates user instructions via two complementary paths:
 - *Requirement-First*: Directly generates complex user intents based on target difficulty.
 - *Tool-Chain-First*: Constructs tool sequences via random walks over API dependency

graphs, then converts them to natural language.

- **EnvironmentAgent**: Synthesizes environment states aligned with instruction complexity.
- **RubricAgent**: Defines task success criteria and constraint verification rules.
- **ValidatorAgent**: Performs format, logic, and quality validation.
- **DeduplicatorAgent**: Removes redundancy via semantic embeddings.

Data Volume and Domain Distribution : (i) **Airline & Retail Domains**. For Airline and Retail domains, we synthesize 1,000 expert trajectories each for SFT training and 250 tasks for RL training respectively. Tasks are generated using the multi-agent framework and stratified by difficulty levels (simple, moderate, complex); (ii) **Telecom Domain**. Due to the dual-control nature of Telecom tasks (requiring agent-user coordination for device operations), we do not synthesize tasks from scratch. Instead, we select high-quality Telecom train tasks from the open-source Tau2 official source (different from test task set) and distill expert trajectories using Claude-4.5-Sonnet for both SFT and RL training.

Pass-Rate-Based Stratification and Our Topological Filtering : After synthesizing the raw corpus, we stratify tasks based on their empirical pass rates using Qwen3-32B-Instruct (without fine-tuning): (i) **SFT Pool (High Pass Rate)**: Tasks with pass rate ≥ 0.7 are classified as *fundamental* and allocated to the SFT pool. For these tasks, we distill expert trajectories using Claude-4.5-Sonnet with few-shot prompting and policy constraints. This yields approximately 3000 candidate trajectories (Airline: 1000, Retail: 1000, Telecom: 1000). TopoCurate then applies topological filtering (Sec. 4.3) to select ~ 2400 high-quality trajectories based on S_{ref} , S_{eff} , and S_{rare} metrics; (ii) **RL Pool (Moderate Pass Rate)**. Tasks with pass rate $\in [0.1, 0.7]$ exhibit sufficient difficulty and are allocated to the RL pool, yielding approximately 600 candidate tasks. Crucially, the final RL task selection depends on the SFT-trained model’s pass rate: We first train an SFT model using the curated SFT data. Then, We re-evaluate the RL task pool using the SFT-trained model to compute updated pass

rates $p_{\text{SFT}}(\mathcal{T})$; TopoCurate then applies topological task selection (Sec. 4.4) within the subset $\{\mathcal{T} : p_{\text{SFT}}(\mathcal{T}) \in [0.1, 0.7]\}$ to select ~ 480 tasks with high structural complexity ($V_{\text{struct}}, V_{\text{div}}$).

Quality Assurance : Each generated sample undergoes three-stage validation: (i) Format Validation: Schema compliance checks; (ii) Logic Validation: Constraint consistency verification; (iii) Quality Validation: Semantic coherence assessment; (iv) Semantic deduplication is performed using embedding-based similarity matching (threshold: cosine similarity > 0.92), reducing redundancy by approximately 12%.

B.2 Training Configurations

Supervised Fine-Tuning Configuration. All experiments are conducted on the Tau2 benchmark environment with mixed-domain training across Airline, Retail, and Telecom tasks. For SFT, we fine-tune the Qwen3-Thinking models using a global batch size of 256. The learning rate is selected from the range $5\text{e-}6, 7\text{e-}6$ via grid search on the validation set. We train for 3 epochs using the AdamW optimizer with weight decay 0.01. The learning rate schedule employs linear warmup for 10% of total steps followed by cosine decay (Wang et al., 2025; Chen et al., 2025b). The mixed-domain training strategy ensures the model learns generalizable tool-use patterns across diverse task distributions.

Reinforcement Learning Configuration. For RL training, we apply Group Relative Policy Optimization (GRPO) on the SFT-trained model using a rollout batch size of 1,024. Each rollout iteration samples 60 tasks in total, with 20 tasks from each domain (Airline, Retail, and Telecom) to maintain balanced exploration. For each sampled task, we generate 16 trajectories to compute group-relative advantages. The reward function is binary sparse: $r = 1$ for task success and $r = 0$ otherwise. We use the AdamW optimizer with a learning rate of $1\text{e-}6$ and train for 80 steps (8 epoch), at which point convergence is observed based on training reward curves.

B.3 Embedding Model Selection for Topology Modeling

We conducted ablation experiments with two open-source embedding models of varying scales across 99 tasks equally sampled from Tau2 domains. Through detailed embedding and statistical analysis of the merging results. Take Jina as the base-

line, we can observe that all three models show high agreement on the Tier construction (merging feature) and the final decision of "which trajectories should be selected as training data". These results can consistently verify the robustness of our method.

Table 5: **Embedding Model Selection.** Comparison of different embedding models used in the Tier construction phase. $\dagger$ represents the correlation coefficient between the selected trajectories and the ground truth ranking.

Model	Params	Dim	Context	Tier Agreement	Spearman $\rho^\dagger$
Jina-v2-base-en	137M	768	8,192	—	—
Owen3-Embedding-0.6B	600M	1,024	32,768	87.8%	0.945

B.4 Sensitivity Analysis of Metric Weights

During the trajectory selection phase, we conducted a systematic sensitivity analysis on the weights of the selection metrics. As presented in Table 6, where our default weight configuration is $\alpha = 0.4, \beta = 0.3, \gamma = 0.3$, we introduced perturbations to the weights to evaluate performance stability. The comparison reveals that the set of selected trajectories and key statistical metrics remained largely stable across different settings. The patterns of trajectory retention/exclusion and the overall selection outcomes exhibited only minor fluctuations without affecting the primary findings. These results demonstrate that the proposed trajectory selection method exhibits strong robustness to weight settings and does not depend on fine-grained weight tuning to achieve stable performance.

Table 6: **Sensitivity Analysis of Trajectory Selection Metric Weights.** The weight settings are denoted as (α, β, γ) , corresponding to the weights for Efficiency, Diversity, and Reflection metrics, respectively.

Weight Settings (α, β, γ)	Airline	Retail	Telecom	Avg
(0.4, 0.3, 0.3)	0.445	0.529	0.435	0.470
(0.4, 0.2, 0.4)	0.413	0.549	0.420	0.461
(0.3, 0.4, 0.4)	0.415	0.517	0.424	0.452

B.5 RL Training Dynamics.

To validate topological curation’s effectiveness across the training pipeline, we conduct two controlled experiments on Qwen3-32B-Thinking, isolating the impact of (i) SFT initialization and (ii) RL task selection. **Impact of SFT Initialization Quality.** We compare two RL runs starting from different SFT checkpoints: one trained on

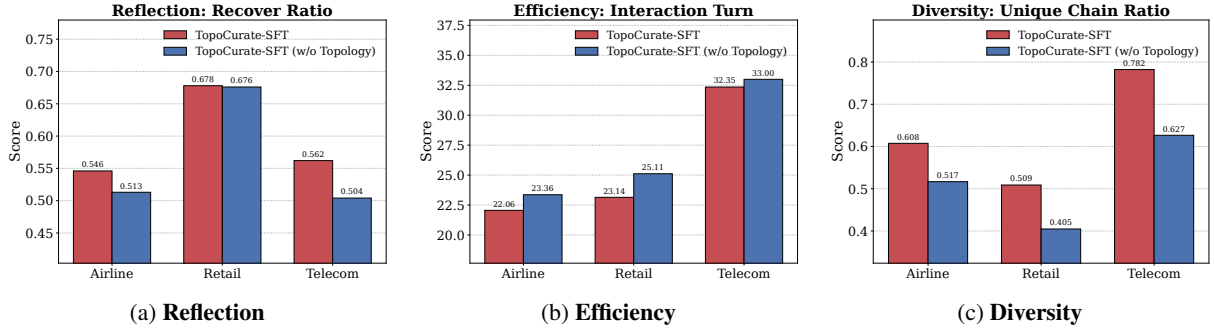


Figure 3: **Model Behavior Analysis.** We compare the behavioral patterns of TopoCurate-SFT (Ours) vs. the baseline SFT (w/o Topology). (a) Ours demonstrates significantly higher reflective recovery rates, particularly in the complex Telecom domain. (b) In terms of the number of interaction turns (lower is better), Ours achieves higher efficiency overall, while maintaining success rate across all domains. (c) Ours consistently generates more diverse tool chains (Unique Chain Ratio), mitigating mode collapse.

TopoCurate-selected trajectories (w/ Topology) and another on outcome-filtered data (w/o Topology), using the same RL task pool. As shown in Figure 5 (left), the topologically-initialized model exhibits significantly higher starting training reward (0.29 vs. 0.22 at step 0) and converges to a superior plateau (0.64 vs. 0.61 at step 60), demonstrating that high-quality SFT provides a stronger foundation for policy optimization. Evaluation accuracy on Tau2 (Figure 4, left) corroborates this: the w/ Topology model consistently outperforms the baseline across all domain; **Impact of RL Task Selection.** To isolate the effect of task selection, we train two models from the same SFT checkpoint (ensuring identical initial training rewards of ~ 0.25) but vary the RL task pool: one selected via topological metrics (V_{struct} , V_{div}) and another via uniform sampling. Figure 5 (right) reveals that while both start identically, topologically-selected tasks yield steeper learning curves and higher final rewards. Evaluation accuracy (Figure 4, right) confirms this: the w/ Topology model achieves consistent gains across all domains, with Telecom showing substantial improvement, validating that topological task selection maximizes gradient SNR.

These experiments establish a causal chain: topological SFT curation $\rightarrow$ robust initialization $\rightarrow$ improved RL starting point; topological task selection $\rightarrow$ high-SNR gradients $\rightarrow$ accelerated convergence. Together, they demonstrate TopoCurate’s compositional impact—both stages contribute independently and synergistically to final performance.

B.6 More Evaluation on BFCLv4

We further evaluated on BFCL v4 as shown in Table 7, a comprehensive benchmark covering

single/multi-turn function calling, live API invocation, agentic memory tasks, and hallucination detection. Results on Qwen3-32B can further verify the effectiveness of our method.

Table 7: **BFCL v4 Results on Qwen3-32B.** Evaluation across multiple sub-tasks. The best result in each column is highlighted in bold.

Method	NON_LIVE	LIVE	MULTI_TURN	AGENTIC	HALLUCINATION
Qwen3-32B-Thinking (Base)	0.475	0.700	0.215	0.197	0.724
TopoCurate-SFT (w/o Topology)	0.531	0.799	0.253	0.201	0.703
TopoCurate-SFT	0.550	0.790	0.261	0.227	0.713
TopoCurate-RL (w/o Topology)	0.539	0.798	0.301	0.255	0.783
TopoCurate-RL	0.562	0.805	0.325	0.261	0.803

B.7 Topology Construction Statistics.

We agree that detailed topology statistics are essential for assessing whether the quotient topology is dense enough to support reliable Success Potential estimation. Table 8 reports the construction statistics across all three Tau2 domains, computed over the full RL candidate pool with 16 rollouts per task and default thresholds $\delta_{\text{tool}} = 0.92$ and $\delta_{\text{result}} = 0.88$.

Table 8: **Topology construction statistics across Tau2 domains.** We report the average number of rollouts per task, raw turns before merging, merged nodes after quotient construction, and the resulting compression ratio.

Domain	Avg Rollouts/Task	Raw Turns (Mean)	Merged Nodes (Mean)	Compression Ratio
Airline	16	392	87	4.5×
Retail	16	428	102	4.2×
Telecom	16	526	138	3.8×

The average number of rollouts per task is set to $K = 16$, establishing the baseline sample volume to ensure sufficient coverage of diverse decision pathways, including both success and failure branches. The raw turns reflect the cumulative number of interaction steps across all rollouts before

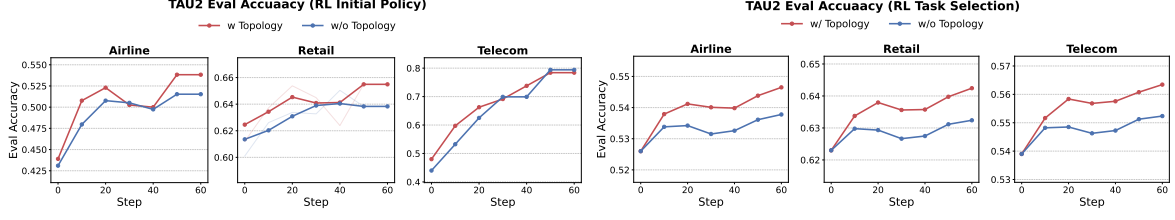


Figure 4: **Evaluation Accuracy on Tau2 for RL.** (Left) Impact of SFT initialization quality on downstream RL performance across Airline, Retail, and Telecom domains. Models initialized with topological SFT exhibit consistently higher accuracy. (Right) Impact of RL task selection on final performance. Topologically-selected tasks lead to superior convergence across all domains, with the most pronounced gains in Telecom.

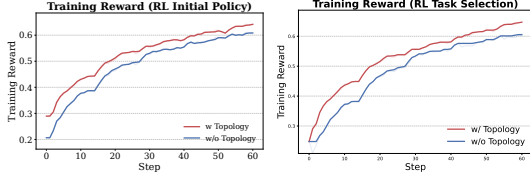


Figure 5: **Training Reward Curves for RL.** (Left) Comparison of models initialized with topologically-curated SFT (w/ Topology) vs. outcome-filtered SFT (w/o Topology), using the same RL task pool. (Right) Comparison of models trained on topologically-selected RL tasks (w/ Topology) vs. uniformly-sampled tasks (w/o Topology) from the same SFT checkpoint. Topological curation improves both initialization quality and task selection efficiency.

merging, and the relatively high counts (392–526) highlight the long-horizon complexity of the multi-turn environments. The merged nodes denote the cardinality of the unique state space after applying the semantic equivalence relation in Eq. (3), where each node represents a semantically distinct logical state, such as a specific API milestone or environmental response. The compression ratio, computed as raw turns divided by merged nodes, ranges from $3.8\times$ to $4.5\times$, indicating that the quotient topology removes redundant behaviors while preserving critical, parameter-sensitive decision states. This balance supports reliable estimation of the Success Potential Field $\Phi(v)$.

These statistics confirm that the topology achieves sufficient densification for $\Phi(v)$ estimation without over-merging semantically distinct states. The conservative compression ratios reflect the dual-constraint design in Eq. (3), which preserves important distinctions such as different API parameters. More importantly, the visitation statistics show a clear practical pattern: a subset of semantically central nodes, such as user verification, product search, or order checkout, is revisited multiple times across different rollouts and thus

accumulates sufficient empirical evidence. Meanwhile, sparse leaf nodes remain low-count and are handled conservatively in downstream scoring, primarily contributing to the diversity-oriented term S_{rare} . This is the regime in which empirical Success Potential estimation becomes meaningful. We will add this table and the accompanying visitation analysis to Section 5.5 in the revision.

B.8 Comparison with Curriculum RL / DUMP.

TopoCurate is a structural refinement on top of pass-rate selection and is complementary to curricula such as DUMP. Our RL pool is already restricted to the medium-difficulty band $p_{\text{SFT}} \in [0.1, 0.7]$ (Appendix B.1), which is the regime curriculum methods typically target, and TopoCurate adds structural filtering within this band.

The key motivation is the *Outcome Equivalence Illusion*: tasks with the same pass rate or advantage magnitude can still have very different training value, since advantage is only a scalar variance signal and cannot capture genuine bifurcation or solution-path diversity. To validate this, we added a DUMP-style advantage-UCB baseline. Table 9 reports the Tau2 Pass@1 results on Qwen3-32B.

Table 9: **Comparison with curriculum RL / DUMP on Qwen3-32B (Tau2 Pass@1).** The best result in each column is highlighted in bold.

Method (RL task selection)	Airline	Retail	Telecom	Avg
w/o Topology (uniform / pass-rate)	0.531	0.643	0.721	0.632
DUMP (advantage-UCB curriculum)	0.540	0.656	0.744	0.647
TopoCurate-RL (Ours)	0.558	0.687	0.784	0.676

TopoCurate-RL achieves 0.676 versus 0.647 for DUMP, showing that structural selection adds gains beyond difficulty-based curriculum alone. This result supports the claim that topology-aware filtering provides additional benefits beyond curriculum learning based only on task difficulty.

B.9 Robustness to Extremely Low Selection Ratios.

We further evaluated TopoCurate under extremely low selection ratios of 5%, 10%, and 20% for SFT on Qwen3-8B. For a fair comparison, we keep the amount of training data the same between the Outcome-based method and TopoCurate on the Tau2 benchmark. Table 10 summarizes the results.

Table 10: **Robustness of TopoCurate under low selection ratios on Qwen3-8B.** The best result in each row is highlighted in bold.

Selection Ratio	Outcome-based	TopoCurate-SFT	Relative Gain
5%	0.312	0.389	+24.7%
10%	0.368	0.421	+14.4%
20%	0.405	0.448	+10.6%

The reported results verify that TopoCurate consistently outperforms the Outcome-based method across all tested selection ratios. This demonstrates that TopoCurate remains robust even when only a small fraction of the data is retained. We will add this analysis to Section 5.3 in the revision.

C More Details about Method

This appendix provides formal theoretical justifications for TopoCurate’s trajectory and task selection mechanisms, grounding our empirical design choices in learning-theoretic principles.

C.1 Computational Complexity of Topology Construction.

Constructing the quotient topology involves pairwise comparisons of interaction turns. A naive implementation scales quadratically $O(N_{\text{turn}}^2)$. To ensure scalability, we employ *Locality-Sensitive Hashing (LSH)* on the multi-view embeddings to retrieve candidate equivalent turns, reducing the complexity to $O(N_{\text{turn}} \log N_{\text{turn}})$. Furthermore, since topological curation is an offline pre-processing step, it incurs zero computational overhead during the actual training loop.

Time Cost Analysis. We acknowledge that constructing a semantic quotient topology will introduce an additional processing stage compared with standard outcome-based filtering. **But the extra time cost during data selection could be acceptable or meaningful in some way as the following table shows.**

Take the task selection for rl training for example, previous pass-rate-selected (outcome-based filtering) strategy also needs the process of generating

Table 11: Time consumption comparison for topology construction and filtering on three domains.

	Telecom	Retail	Airline
Construction_Topology	38.3 min	33.3 min	40.1 min
Outcome_Reward_Filtering	300.5 min	233.6 min	227.2 min

multi-trial rollouts, while we just further provide a simple but effective topology analysis to evaluate the training potential of training tasks. We just introduce extra 10%-30% data selection time, the rl task set could also be reduced to 75% compared with outcome filtering, which means will reduces RL training time by 30% while achieving superior training performance (6.9% performance gain).

C.2 Mitigate the Covariate Shift and Mode Collaps through Trajectory Selection

Our mechanism can be viewed as a generalized Rejection Sampling scheme. While standard Outcome Filtering assumes a uniform prior ($p(\tau) \propto \mathbb{I}[R = 1]$), TopoCurate modulates this prior using the topological potential:

$$p(\tau) \propto \mathbb{I}[R = 1] \cdot \exp(w(\tau)) \quad (11)$$

This effectively shifts the sampling distribution towards the process quality, suppressing sub-optimal lucky guesses. Formally, SFT minimizes the negative log-likelihood $\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{\tau \sim p(\tau)}[\log \pi_{\theta}(\tau)]$. By reweighting the data distribution, our method explicitly minimizes the KL divergence $D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{expert}})$ between the learned policy and an ideal robust expert that balances recovery, efficiency, and diversity. This realignment mitigates Covariate Shift (by including error-recovery patterns) and Mode Collapse (by enforcing diversity), addressing the dual deficits of outcome filtering.

Problem Formulation: SFT as Distribution Matching. Supervised Fine-Tuning minimizes the forward KL divergence between the policy π_{θ} and the empirical data distribution $\hat{p}_{\text{data}}$:

$$\begin{aligned} \mathcal{L}_{\text{SFT}}(\theta) &= \mathbb{E}_{\tau \sim \hat{p}_{\text{data}}}[-\log \pi_{\theta}(\tau)] \\ &= D_{\text{KL}}(\hat{p}_{\text{data}} \parallel \pi_{\theta}) + \text{const.} \end{aligned} \quad (12)$$

Standard outcome filtering defines $\hat{p}_{\text{data}}(\tau) \propto \mathbb{I}[R(\tau) = 1]$, implicitly assuming all successful trajectories are equally valuable. This uniform weighting ignores critical distinctions in process quality, leading to three pathologies:

Pathology 1: Covariate Shift from Fragile Trajectories. Consider a task with two solution paths: τ_{fragile} succeeds via a lucky sequence that avoids constraint checking, while τ_{robust} explicitly validates constraints before proceeding. Standard filtering assigns equal weight: $\hat{p}(\tau_{\text{fragile}}) = \hat{p}(\tau_{\text{robust}})$. However, at test time, environment perturbations (e.g., stricter error handling) render τ_{fragile} invalid, causing distribution shift. Formally, let p_{test} denote the test distribution. The expected test loss suffers from:

$$\mathbb{E}_{\tau \sim p_{\text{test}}} [\ell(\pi_\theta, \tau)] \geq \mathbb{E}_{\tau \sim \hat{p}_{\text{data}}} [\ell(\pi_\theta, \tau)] + D_{\text{TV}}(p_{\text{test}}, \hat{p}_{\text{data}}) \quad (13)$$

where D_{TV} is the total variation distance. By prioritizing τ_{robust} via Reflective Recovery (S_{ref}), TopoCurate explicitly reduces $D_{\text{TV}}(p_{\text{test}}, \hat{p}_{\text{data}})$ by aligning the training distribution with robust behaviors that generalize under perturbations.

Pathology 2: Token Inefficiency from Redundant Loops. Let τ_{verbose} contain redundant API calls (e.g., querying the same flight ID thrice) while τ_{concise} achieves the same outcome efficiently. Standard filtering weighs them equally, but inference cost scales linearly with trajectory length. The expected inference cost is:

$$C_{\text{infer}} = \mathbb{E}_{\tau \sim \hat{p}_{\text{data}}} [|\tau|] \cdot c_{\text{token}} \quad (14)$$

where c_{token} is the per-token cost. By penalizing τ_{verbose} via Semantic Efficiency (S_{eff}), TopoCurate minimizes $\mathbb{E}[|\tau|]$, directly optimizing for token economy. Empirically, this reduces inference latency by up to 20% (see Sec. 5.3).

Pathology 3: Mode Collapse from Solution Homogeneity. Suppose 80% of successful trajectories follow a dominant path τ_{common} , while 20% explore alternative strategies $\{\tau_{\text{rare}, i}\}$. Uniform weighting biases the policy toward τ_{common} , suppressing exploration. This manifests as mode collapse: the policy fails on tasks requiring τ_{rare} . Formally, the entropy of the learned policy is bounded by:

$$H(\pi_\theta) \leq H(\hat{p}_{\text{data}}) + \log N_{\text{support}} \quad (15)$$

where N_{support} is the effective support size of $\hat{p}_{\text{data}}$. By upweighting τ_{rare} via Distributional Diversity (S_{rare}), TopoCurate increases N_{support} , maximizing $H(\pi_\theta)$ and mitigating mode collapse.

Optimality of Weighted Sampling. TopoCurate reweights the data distribution as:

$$\hat{p}_{\text{TopoCurate}}(\tau) \propto \mathbb{I}[R(\tau) = 1] \cdot \exp(w(\tau)) \quad (16)$$

where $w(\tau) = \lambda_{\text{ref}} S_{\text{ref}}(\tau) + \lambda_{\text{eff}} S_{\text{eff}}(\tau) + \lambda_{\text{rare}} S_{\text{rare}}(\tau)$. This can be viewed as tilted importance sampling with target distribution $p^*(\tau) \propto p_{\text{expert}}(\tau)$, where p_{expert} is a hypothetical robust expert policy. The KL divergence between the learned policy and the expert is:

$$D_{\text{KL}}(\pi_\theta \| p_{\text{expert}}) = \mathbb{E}_{\tau \sim p_{\text{expert}}} [\log p_{\text{expert}}(\tau)] - \mathbb{E}_{\tau \sim p_{\text{expert}}} [\log \pi_\theta(\tau)] \quad (17)$$

By aligning $\hat{p}_{\text{TopoCurate}}$ with p_{expert} , TopoCurate directly minimizes this divergence, ensuring the policy internalizes robust, efficient, and diverse strategies.

C.3 Maximizing Signal-to-Noise Ratio through Task Selection

In GRPO, the policy gradient’s informativeness is determined by the variance of advantages $\text{Var}[\hat{A}]$. Tasks yielding homogeneous rollouts (all succeed or all fail) produce $\text{Var}[\hat{A}] \approx 0$, resulting in vanishing gradient updates. Our structural metrics V_{struct} and V_{div} serve as tractable proxies for the Signal-to-Noise Ratio (SNR) of the gradient: tasks with high V_{struct} exhibit sharp decision boundaries, leading to large advantage gaps and high-variance gradients. Formally, the Fisher information matrix $\mathcal{I}(\theta) = \mathbb{E}[\nabla \log \pi \nabla \log \pi^\top]$ quantifies the sensitivity of the policy to parameter changes. Tasks with high structural variance $\mathbb{E}[\text{Var}(\Phi)]$ maximize $\mathcal{I}(\theta)$, ensuring each gradient update carries maximal information about the optimal policy. Thus, our task selection directly maximizes gradient SNR by prioritizing tasks where outcomes significantly diverge based on critical actions, effectively addressing the fundamental vanishing signal problem in sparse-reward settings.

Problem Formulation: RL Gradient Informativeness. In sparse-reward RL with Group Relative Policy Optimization (GRPO), the policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim P_{\text{task}}} \mathbb{E}_{\{\tau_k\}_{k=1}^K \sim \pi_\theta(\cdot | \tau)} \left[\sum_{k=1}^K \nabla_\theta \log \pi_\theta(\tau_k) \cdot \hat{A}(\tau_k) \right] \quad (18)$$

where $\hat{A}(\tau_k) = r(\tau_k) - \bar{r}$ is the group-relative advantage. The gradient’s Signal-to-Noise Ratio (SNR) is:

$$\text{SNR} = \frac{\|\mathbb{E}[\nabla_{\theta} J]\|^2}{\text{Var}[\nabla_{\theta} J]} \quad (19)$$

Standard task selection (e.g., uniform sampling or pass-rate thresholds) fails to account for the structural properties of tasks that determine SNR, leading to two failure modes:

Failure Mode 1: Vanishing Gradients from Homogeneous Rollouts. Consider a task $\mathcal{T}_{\text{trivial}}$ where all K rollouts yield identical outcomes (either all succeed or all fail). The advantage variance is:

$$\text{Var}[\hat{A}] = \frac{1}{K} \sum_{k=1}^K (\hat{A}(\tau_k) - 0)^2 = 0 \quad (20)$$

since $\hat{A}(\tau_k) = r - \bar{r} = 0$ for all k . Consequently, $\nabla_{\theta} J \approx 0$, providing no learning signal. The expected gradient magnitude is bounded by:

$$\|\mathbb{E}[\nabla_{\theta} J]\| \leq \|\nabla_{\theta} \log \pi\| \cdot \text{Var}[\hat{A}]^{1/2} \quad (21)$$

For tasks with $\text{Var}[\hat{A}] \approx 0$, the gradient vanishes regardless of the policy’s capacity. TopoCurate avoids such tasks by prioritizing high Error Branch Ratio (V_{struct}), ensuring $\text{Var}[\hat{A}] \gg 0$.

Failure Mode 2: Low Gradient Variance from Mode Collapse. Consider a task $\mathcal{T}_{\text{single-path}}$ solvable via only one tool sequence. All successful rollouts follow the same path, leading to low policy entropy:

$$H(\pi_{\theta}|\mathcal{T}) = - \sum_{\tau} \pi_{\theta}(\tau|\mathcal{T}) \log \pi_{\theta}(\tau|\mathcal{T}) \rightarrow 0 \quad (22)$$

Low entropy implies the policy has converged prematurely to a single mode, suppressing exploration. The Fisher information matrix $\mathcal{I}(\theta)$ quantifies the sensitivity of the policy:

$$\mathcal{I}(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau)^{\top}] \quad (23)$$

For single-mode distributions, $\mathcal{I}(\theta)$ has low rank, limiting gradient diversity. TopoCurate selects tasks with high Strategic Heterogeneity (V_{div}), ensuring $\mathcal{I}(\theta)$ remains full-rank and gradient updates explore diverse directions.

Optimality of Structural Metrics. TopoCurate defines the task sampling distribution as:

$$P_{\text{select}}(\mathcal{T}) \propto \exp \left(\frac{V_{\text{struct}}(\mathcal{T}) + \alpha V_{\text{div}}(\mathcal{T})}{T} \right) \quad (24)$$

This prioritizes tasks where:

- High V_{struct} ensures sharp decision boundaries, maximizing $\text{Var}[\hat{A}]$ and gradient magnitude.
- High V_{div} ensures multiple valid solution paths, maximizing $\text{rank}(\mathcal{I}(\theta))$ and preventing premature convergence.

The expected SNR under TopoCurate’s distribution is:

$$\mathbb{E}_{\mathcal{T} \sim P_{\text{select}}} [\text{SNR}] \propto \mathbb{E}[V_{\text{struct}}] \cdot \mathbb{E}[V_{\text{div}}] \quad (25)$$

Maximizing this expectation directly optimizes for gradient informativeness.

Connection to Fisher Information and Natural Gradients. The Fisher information matrix $\mathcal{I}(\theta)$ defines the natural gradient:

$$\tilde{\nabla}_{\theta} J = \mathcal{I}(\theta)^{-1} \nabla_{\theta} J \quad (26)$$

Natural gradients converge faster than standard gradients by accounting for the curvature of the policy manifold. Tasks with high V_{struct} and V_{div} increase the smallest eigenvalue $\lambda_{\min}(\mathcal{I}(\theta))$, reducing the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ and accelerating convergence. Formally:

$$\|\theta^{(t+1)} - \theta^*\| \leq \left(1 - \frac{\eta}{\kappa}\right) \|\theta^{(t)} - \theta^*\| \quad (27)$$

where θ^* is the optimal policy. By improving κ , TopoCurate accelerates RL training.

Sample Efficiency Bounds. For standard task selection with uniform sampling, the sample complexity to achieve ϵ -optimal policy is:

$$N_{\text{tasks}} = O \left(\frac{|\mathcal{A}|^2 |\mathcal{S}|^2}{\epsilon^2 (1 - \gamma)^4} \right) \quad (28)$$

where $|\mathcal{A}|$, $|\mathcal{S}|$ are the action and state space sizes, and γ is the discount factor. TopoCurate reduces this by a factor proportional to the average SNR:

$$N_{\text{tasks}}^{\text{TopoCurate}} = O \left(\frac{|\mathcal{A}|^2 |\mathcal{S}|^2}{\epsilon^2 (1 - \gamma)^4 \cdot \mathbb{E}[\text{SNR}]} \right) \quad (29)$$

Robustness to Task Distribution Shift. In deployment, the test task distribution $p_{\text{test}}(\mathcal{T})$ may differ from training. Tasks with high structural diversity (high V_{div}) provide better generalization. Formally, the generalization gap is bounded by:

$$\begin{aligned} \mathbb{E}_{\mathcal{T} \sim p_{\text{test}}} [J(\theta)] - \mathbb{E}_{\mathcal{T} \sim p_{\text{train}}} [J(\theta)] \\ \leq C \cdot D_{\text{KL}}(p_{\text{test}} \parallel p_{\text{train}}) \end{aligned} \quad (30)$$

TopoCurate minimizes this gap by training on structurally diverse tasks, reducing the effective D_{KL} through improved coverage of the task manifold.

Robustness in Low Data Regimes: In scenarios where trajectory data is sparse, the topological structure may be underdeveloped. In such cases, TopoCurate naturally degrades to a uniform sampling strategy (as $\Phi(v)$ becomes uniform), providing a *safe fallback mechanism*.

D Case Studies

D.1 Trajectory Comparison for Agent Behaviours

At the trajectory level, we compare multiple successful trajectories within the same task. Regarding diversity, different correct trajectories for the same task exhibit distinct step-by-step procedures and reasoning paths, such as what information is checked first, how the problem is decomposed, and how the agent transitions between subtasks. Regarding efficiency, we contrast pairs of trajectories that share the same starting conditions and reach the same successful outcome. Despite achieving identical results, they differ in their intermediate steps and overall length, reflecting distinct ways of organizing the process like using a single consolidated processing loop versus a phased multi-turn progression.

D.2 Task Comparison for Training Potential Evaluation

At the task level, we observe that even when two tasks have a similar number of correct trajectories or comparable success rates, they may still show significant differences in the number of valid Solution Families they contain. This reveals the varying degrees of convergence or diversity manifested in the overall structure of the tasks.

E Ethics and Resource Statement

Dataset. Our experiments use public benchmarks including Tau2Bench, BFCLv3 and BFCLv4. Synthetic task trajectories are adapted from the APIGen-MT pipeline on the Tau2 environment. No human annotations are collected.

Models. We use Qwen3-8B / Qwen3-32B-Thinking as our backbone model. Claude-4.5-Sonnet is used for distilling seed expert trajectories. Embedding models include Jina-embeddings-v2-base-en and Qwen3-0.6B-Embedding.

LLM Usage. Large-language models are used for paper writing polishing only and not for generating core experimental results.

Computation. Experiments are conducted on GPU clusters for SFT and GRPO reinforcement-learning training.

Trajectory Comparison | Diversity

Task:	Change the existing flight/reservation from JFK to LAX to a flight to the Bay Area (SFO or OAK), and strongly prefers not to cancel and rebook.
Trajectory 1:	Treat the issue as a "rules/system boundaries" problem. First, clearly explain whether the destination can be changed, and then move on to the next steps.
Typical steps:	Collect identity and order information → Confirm facts/background → Clarify hard constraints → Provide actionable solutions within the constraints
Trajectory 2:	Quickly reject infeasible requests, then immediately provide the information needed for the user's decision (flights/prices).
Typical steps:	Collect identity and order information → Provide a quick conclusion → Immediately present alternative solutions → Give guidance for the next step
Comparison:	Trajectory 1: First, explain the rules thoroughly, then proceed. Trajectory 2: Directly give the conclusion, then provide the alternative flight prices.

Trajectory Comparison | Efficiency

Task:	The user needs to modify the pending order, changing the T-shirt size from M to L and the shoe color from black to blue. Confirm the price difference and use the \$75 gift card balance to cover any extra cost.
Current State:	The user wants to modify the pending order and has completed email verification. The next step is to locate the order and process the changes and price difference payment.
Trajectory 1:	After email verification, merge the process for a single cycle, shorter.
Typical steps:	Email verification → Locate pending order → Check product variants / calculate price difference → Confirm gift card balance can cover → User confirmation → Submit modification and return success result
Trajectory 2:	After email verification, proceed in phases, with additional calculation steps, resulting in multiple cycles, longer.
Typical steps:	Email verification → Output order status / summary and confirm pending → User clarifies modification request → Check product variants + additional price difference calculation → Confirm gift card availability → User confirmation → Submit modification and return success result
Comparison:	With the same user introduction and goal, Trajectory 1: Combine processing (locating the order and addressing modification requests in one step), making it shorter. Trajectory 2: Use "phased processing (first confirming the order, then having the user restate their request) + additional calculation steps," making it longer.

Task Comparison

Task 1:	The user is unable to send MMS (multimedia messages) and needs to troubleshoot and restore functionality.
Solution type:	Type 1: Connectivity → permissions closure: restore connectivity first (enable mobile data, reseal SIM, etc.), then identify missing SMS permission and enable it to fix MMS. Type 2: SIM-first → permissions convergence: fix SIM/no-signal issues first, then enable missing SMS permission to resolve MMS.
Task 2:	The user is in France and experiencing slow or intermittent mobile data, requesting "excellent speed."
Solution type:	Type 1: Usage/limit-first + settings optimization: refuel/add data first, then enable data roaming and disable Data Saver, reaching "excellent" speed in a speed test. Type 2: Settings-first (roaming → data saver): enable Data Roaming first, then disable Data Saver to improve speed. Type 3: Settings-first (data saver → roaming): disable Data Saver first, then enable Data Roaming. Type 4: Recovery/backtracking type: after a setting change causes loss of connectivity or user anxiety, proceed with step-by-step checks/restart, and still end up with the combo fix.
Comparison:	Task 1: 8 correct trajectories, 2 types, more "convergent" (most correct trajectories fall into a few ending shapes). Task 2: 8 correct trajectories, 4 types, ending shapes are more dispersed (more types).