ACE-SAMPLER: HIERARCHICAL STOCHASTIC DATA SELECTION FOR TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training tool-using language models increasingly relies on heterogeneous collections of agent trajectories, yet larger data pools can introduce redundancy, imbalance, and unreliable supervision. Selecting useful agentic data requires capturing state-dependent behaviors and accounting for the uneven learning value of turns within a trajectory. We introduce ACE-SAMPLER, a hierarchical stochastic sampler that jointly considers **A**ccuracy, **C**omplexity, and div**E**rsity under a fixed turn-level training budget. At its core is a factorized behavioral representation that captures interaction patterns and action dependencies beyond surface semantics. Using this representation, ACE-SAMPLER first samples trajectories for behavioral coverage, then prioritizes important assistant turns and samples them for novelty. A stochastic continuation rule balances coverage across trajectories with supervision within each trajectory, while validity checks and learner-aware complexity estimation favor valid examples of appropriate complexity. Across three tool-use benchmarks and three model scales, ACE-SAMPLER consistently outperforms the compared selectors with a budget of 5,000 training turns, improving aggregate accuracy over random selection by more than 4 percentage points on average. Ablations and scaling experiments examine the effects of behavioral representations, validity filtering, learner-aware complexity, and training data budgets.

1 INTRODUCTION

Large Language Model (LLM)-based agentic systems are increasingly used to solve tasks that require more than a self-contained text response. By iteratively interacting with external tools, LLM agents can retrieve information, operate software, and complete long-horizon tasks (Chen et al., 2024b; Barres et al., 2025; Li et al., 2026). This interaction pattern has made tool use a central capability for post-training LLMs (Qin et al., 2024; Liu et al., 2025). The value of the training experience therefore depends not only on the final answer, but also on whether the model learns to make valid, well-timed, and state-dependent decisions throughout an interaction (Prabhakar et al., 2025; Song et al., 2026).

The rapid growth of agent research has produced a large and heterogeneous collection of open-source agentic datasets (Liu et al., 2025; Prabhakar et al., 2025; Xu et al., 2025; Yang et al., 2025b; Xu et al., 2026b). These datasets differ in their domains, tool interfaces, task objectives, and trajectory lengths. In practical training pipelines, data may also arrive from different producers, simulators, or scaffolds, each contributing examples with different distributions and quality levels (Xu et al., 2026a). Simply pooling all available trajectories increases the amount of supervision, but can also introduce substantial redundancy and imbalance (Liu et al., 2024b; Dai et al., 2025; Yang et al., 2025c). Under a fixed training budget, frequently occurring interaction patterns can crowd out less common but informative behaviors. This problem is particularly pronounced for agentic data: long trajectories are costly to train on, and useful decisions may coexist with redundant steps, erroneous actions, or structural inconsistencies. Indiscriminate training can therefore waste computation and reinforce undesirable behaviors. Data selection must decide both which experiences to retain (Liu et al., 2024b; Xia et al., 2024; Li et al., 2024b; Zhang et al., 2025) and which assistant turns within those experiences provide useful supervision (Chen et al., 2025c).

Effective selection for agentic data presents two closely related challenges. First, diversity must reflect interaction behavior rather than surface semantics alone. For conventional instruction-response pairs, semantic variation in the instruction often provides a useful proxy for diversity (Liu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2024b; Zhang et al., 2025; Yang et al., 2025c). For agents, however, similar task descriptions may require different actions because of different initial states, while linguistically different trajectories may express the same behavior (Prabhakar et al., 2025; Song et al., 2026; Chen et al., 2026). Second, selection must connect trajectory-level coverage with turn-level learning value. Selecting entire trajectories treats their constituent turns uniformly, whereas selecting turns independently can overlook the broader behavioral patterns in which they occur. A useful selector must preserve coverage across trajectories while prioritizing informative decisions within them, accounting for both sample validity and difficulty relative to the learner (Li et al., 2024b; Zeng et al., 2026).

We introduce ACE-SAMPLER, a hierarchical stochastic data selection framework for heterogeneous tool-use data that jointly considers **A**ccuracy, **C**omplexity, and div**E**rsity. At its core is a factorized behavioral representation that connects trajectory-level interaction patterns with turn-level decisions. Combining direct extraction with offline LLM-assisted annotation, we characterize trajectories by their scenarios, task types, tools, dependency patterns, and sequences of turn functions; individual turns are further described by their behavioral roles, interaction relations, and tool-call structures. These factors make behaviorally relevant distinctions explicit and provide a shared basis for measuring redundancy across heterogeneous sources. To promote behavioral diversity, ACE-SAMPLER first admits trajectories according to their novelty, then considers assistant turns in order of importance and samples them using both trajectory- and turn-level factors. A stochastic continuation rule limits repeated selection from the same trajectory, balancing coverage across trajectories with supervision within each trajectory. For accuracy, rule-based checks filter out structurally invalid samples. For complexity, a learner-aware difficulty score (Zeng et al., 2026) favors examples at an intermediate difficulty for the target model. The resulting procedure allocates a fixed turn-level training budget without requiring access to the original execution environments.

We evaluate ACE-SAMPLER on three representative tool-use benchmarks (BFCL (Patil et al., 2025), τ^2-Bench (Barres et al., 2025), and ACEBench (Chen et al., 2025a)) using a candidate pool of 21,000 trajectories and approximately 180,000 assistant turns from seven heterogeneous sources. Under matched selection budgets and training settings, ACE-SAMPLER achieves the strongest aggregate performance among the compared selectors across Qwen3 models. With only 5,000 selected training turns, it improves aggregate accuracy over random selection by over 4% in average in absolute terms, and even outperforms traning on the full candidate pool in BFCL Multi-Turn. Ablations and data-scaling experiments further examine behavioral representations, validity filtering, learner-aware complexity, and the effect of the selection budget.

Our contributions include:

- We develop a factorized representation of agent behavior at both the trajectory and turn levels, enabling diversity estimation beyond surface semantics.
- We introduce ACE-SAMPLER, a hierarchical stochastic selector that coordinates behavioral coverage across trajectories with importance-aware turn selection, incorporating validity and learner-aware complexity under a fixed training budget.
- We provide an empirical study across multiple benchmarks, model scales, selection baselines, ablations, and data budgets, demonstrating the effectiveness of our hierarchical selection.

2 RELATED WORK

Agent Training. Early tool-learning work primarily focuses on teaching models how to interact with external tools, generally through real APIs or simulated tools (Qin et al., 2024; Liu et al., 2024c; 2025). More recent pipelines construct multi-turn and verifiable trajectories using simulated or executable environments (Wang et al., 2026; Dong et al., 2026; Xu et al., 2026b), and use these trajectories for supervised fine-tuning or reinforcement learning. However, these efforts mainly emphasize how to generate and validate trajectories, rather than how to efficiently allocate a limited training budget across them. As an increasing number of heterogeneous agentic datasets become publicly available (Liu et al., 2025; Yang et al., 2025b; Prabhakar et al., 2025; Xu et al., 2025; Song et al., 2026), there is a growing need for methods that can better utilize such diverse data sources.

Data Selection. Data selection for conventional instruction tuning has largely been studied through the trade-off among quality, diversity, and complexity (Zhang et al., 2025). Several works filter

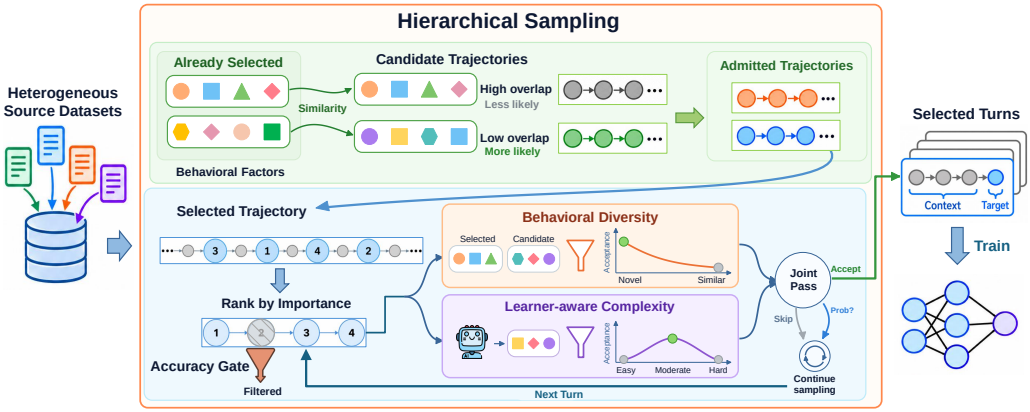

Figure 1: Overview of ACE-SAMPLER. Hierarchical stochastic sampling promotes trajectory-level behavioral coverage and selects importance-ranked assistant turns through behavioral similarity and accuracy and learner-aware complexity gates under a fixed turn-level training budget.

noisy or low-quality instruction-response pairs using stronger evaluators or weak-to-strong model cascades (Chen et al., 2024a; Li et al., 2024a), while others leverage model uncertainty or self-reflection to identify examples that are informative for the target model (Du et al., 2023; Li et al., 2024b; Liu et al., 2024a). Diversity-oriented methods select representative or novel examples using semantic tags, information gain, influence estimation, or gradient similarity (Lu et al., 2024; Chen et al., 2025b; Li et al., 2024c; Xia et al., 2024; Zhao et al., 2025). Complementary studies further show that balanced capability coverage and explicit novelty are important when compressing a large instruction corpus (Dai et al., 2025; Yang et al., 2025c).

When moving to reasoning and agentic settings, these methods may become insufficient, as behavioral diversity and multi-turn interactions are difficult to capture with conventional sample-level representations. Several recent works have explored individual aspects of agentic data quality. For example, ATLaS identifies critical steps within expert trajectories (Chen et al., 2025c), ToolACE-R introduces a model-aware difficulty score (Zeng et al., 2026), and trajectory diversity has also been studied (Chen et al., 2026). Yet these efforts typically address only one aspect of agentic data quality, rather than jointly considering trajectory-level behavioral coverage and turn-level training value under a fixed budget. Our work addresses this gap through hierarchical stochastic selection.

3 HIERARCHICAL STOCHASTIC SAMPLING

3.1 PROBLEM FORMULATION

We consider a pool of complete agent trajectories collected from heterogeneous sources. A trajectory τ is defined as:

$$\tau = (o_0, a_1, o_1, \ldots, a_T, o_T) \tag{1}$$

where a_i ($i \in [1, T]$, T is the total number of assistant turns) are assistant turns recording actions including text responses and tool calls, observations o_j ($j \in [0, T]$) can be user messages or environment feedback, and environment setup (including information like available tools and any policies) is generally included in o_0.

Each trajectory can be divided into T turn-level training examples, with all information before the target assistant turn a_t as context, and a_t as learning target:

$$d(\tau, a_t) = \{(o_0, a_1, o_1, \ldots, a_{t-1}), a_t\}, t \in [1, T] \tag{2}$$

The goal is to select a fixed budget of B turn-level training examples from the trajectory pool, expecting them to cover diverse behaviors and benefit the agent training. Figure 1 draws the overall picture of our hierarchical stochastic sampling method.

3.2 FACTOR-LEVEL ANNOTATION

Diversity in conventional instruction data is often estimated from semantic differences in the instruction text. For agentic data, however, behavior also depends on the interaction state: textually similar trajectories may require different actions, while different surface forms may express the same behavior. Whole-text embeddings may therefore overlook distinctions that matter for learning. We explicitly characterize behavioral factors at both the trajectory and turn levels to support diversity-aware sampling. Below we introduce the factors.

Trajectory-level Factors. We characterize each complete trajectory τ using five factor groups: *scenario*, *task type*, *used tools*, *dependency pattern*, and *turn-function pattern*. Scenario and task type capture the broad domain and objective. Used tools identifies the tool categories involved and is extracted directly from the data without additional annotation. Dependency pattern captures how actions depend on earlier observations or state changes, while turn-function pattern summarizes the sequence of behavioral roles, such as planning, information gathering, execution, recovery, and completion. The latter two factors distinguish trajectories that are textually similar but operationally different, and they can be aggregated based on the below turn-level factors.

Turn-level Factors. We characterize each assistant turn a_t using five behavioral factor groups: *turn function*, *interaction relation*, *target tool names*, *call mode*, and *tool schema*. Turn function describes the turn's role within the trajectory, while interaction relation captures its relationship to earlier user messages or environment feedback. Target tool names identify the tools invoked, call mode describes how the calls are organized, and tool schema captures argument structure and the associated structural complexity. These last three factors are extracted directly from the data without additional annotation. Together, the five groups characterize the behavior supervised at a given interaction state. Inspired by Chen et al. (2025c), we also annotate each turn's *importance* based on its contribution to task completion, allowing critical actions to receive priority during turn selection.

LLM-assisted Annotation. For factors that cannot be extracted directly, we perform an offline LLM-assisted annotation pass over complete trajectories. Given the full ordered interaction, the annotator first assigns trajectory-level labels, then labels each assistant turn and assesses its importance. The structured prompt (Appendix A.3) uses a compact, closed vocabulary for scenarios and the main behavioral categories to encourage consistent annotations.

3.3 DIVERSITY-ORIENTED HIERARCHICAL SELECTION

3.3.1 FACTORIZED BEHAVIORAL SIMILARITY

We first introduce how to estimate similarity based on annotated factorized behaviors. For each factor group g, we represent the annotations of an item x as a set $F_g(x)$. We use the Jaccard similarity:

$$J_g(x, y) = \frac{|F_g(x) \cap F_g(y)|}{|F_g(x) \cup F_g(y)|} \tag{3}$$

The similarity then will be the average over non-empty factor groups with uniform group weights:

$$S(x, y) = \frac{\sum_{g \in \mathcal{G}} J_g(x, y) \mathbf{1}[F_g(x) \cup F_g(y) \neq \emptyset]}{\sum_{g \in \mathcal{G}} \mathbf{1}[F_g(x) \cup F_g(y) \neq \emptyset]}. \tag{4}$$

This group-wise formulation prevents a factor with many labels from dominating the similarity score.

3.3.2 TRAJECTORY-LEVEL STOCHASTIC SELECTION

Let $\mathcal{R}$ denote the set of trajectories that have already contributed at least one selected turn. For a candidate trajectory τ, its similarity to the selected trajectory set is:

$$s_\tau = \max_{\tau' \in \mathcal{R}} S(\tau, \tau') \tag{5}$$

with $s_\tau = 0$ when $\mathcal{R}$ is empty. The trajectory is admitted with probability

$$p_\tau = (1 - s_\tau)^\gamma \tag{6}$$

where γ controls the strength of the diversity preference. Thus, a novel trajectory is likely to be opened, while a trajectory highly similar to an already represented one is less likely to consume the budget. The stochastic gate, rather than a hard novelty constraint, preserves a non-zero chance of sampling useful variants. Once a trajectory contributes a turn, it is finalized for the current pass and its factor representation is added to $\mathcal{R}$.

3.3.3 Turn-Level Selection within a Trajectory

After a trajectory is admitted, we need to determine which turn(s) are most valuable to be learned, especially for those long trajectories with extensive assistant turns. We first follow the trajectory-level selection, define the diversity acceptance probability of each candidate turn t as:

$$p_t^{\text{div}} = (1 - s_t)^{\gamma}. \tag{7}$$

where s_t is maximum similarity to all previous selected turns. Here, when computing similarity, both trajectory and turn factors are considered.

A different point from the trajectory-level selection is that the turns within one trajectory are not totally independent. During stochastic sampling, which item is first placed may significantly influence the final selection. Therefore, we no longer randomly process candidates during turn selection, but sort the candidate turns based on their importance scales, which has been annotated in Section 3.2.

Finally, we make each trajectory contribute at most K turns. After each accepted turn, we decide whether to continue examining turns from the same trajectory. If n candidate turns are available and k turns have already been selected from the trajectory, continuation uses

$$p_{\text{cont}}(n, k) = \frac{n}{n + \eta} \frac{1}{2^k}, \tag{8}$$

where η is a saturation parameter. This favors trajectories with more available supervision while making repeated selection from one trajectory progressively less likely.

3.4 Learner-Aware Complexity Estimation

Behavioral diversity alone does not ensure that a sample is suitable for the current learner. We therefore also consider sample complexity, which depends on the learner's capabilities: the same example may be trivial for one model but challenging for another (Li et al., 2024b; Zeng et al., 2026). Because tool use requires precise function formats and arguments, we adopt the model-aware score $s_t^{\text{dif}} \in [0, 1]$ proposed by Zeng et al. (2026), which measures parameter-level similarity between the model output and the ground-truth tool call, rather than relying on output probability. Intermediate scores indicate partial agreement with the target, suggesting examples that offer a useful learning challenge without being either trivial or excessively difficult. We define the difficulty acceptance probability as

$$p_t^{\text{dif}} = 1 - 2 \left| s_t^{\text{dif}} - c \right|, \tag{9}$$

where $c = 0.5$ maximizes acceptance at intermediate scores and reduces it toward either extreme.

We apply this gate to each target turn that passes the diversity gate, favoring samples that combine behavioral novelty with a difficulty level suited to the current learner.

3.5 Sample Accuracy

Sample accuracy is a basic requirement for effective training, as hallucinations and errors can introduce misleading supervision. We apply a rule-based checker to filter out samples with format errors, hallucinated arguments, or structural inconsistencies. Table 3 shows the dominant error types.

Verifying semantic and logical correctness remains challenging for agentic data, even with LLM assistance, particularly in complex interactions. Recent work has explored more comprehensive verification methods (Prabhakar et al., 2025; Barke et al., 2026). We leave their integration to future work and focus here on evaluating the effectiveness of hierarchical selection.

The complete procedure is summarized in Algorithm 1.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We collect tool-using data from seven data sources: 1) ToolACE (Liu et al., 2025), 2) APIGen-MT (Prabhakar et al., 2025), 3) Toucan (Xu et al., 2025), 4) ToolMind (Yang et al., 2025b), 5) Nemotron Tool Data[1], 6) EnvScaler (Song et al., 2026), and 7) EnvFactory (Xu et al., 2026b). The statistics of these datasets can be found in Table 6. These datasets contain tool-use trajectories spanning diverse domains and environments (including real, synthetic, and constructed executable environments) with varied interaction patterns and levels of complexity, forming rich and diverse candidate pools. To make the selection problem comparable across sources, we sample 3,000 trajectories from each dataset to form the original trajectory pool. The resulting pool contains 21,000 trajectories and approximately 180,000 assistant turns. Unless otherwise specified, the target training budget is $B = 5,000$ turn-level examples. We explore more different budgets in Section 4.2.3.

Evaluation. We evaluate the performance in three representative benchmarks: BFCL (Patil et al., 2025), τ^2-Bench (Barres et al., 2025), and ACEBench (Chen et al., 2025a). For τ^2-Bench, we evaluate the Retail and Airline subsets; and we focus on Agent category in ACEBench. All the evaluations report accuracy score, evaluating how many tasks are successfully completed. For BFCL Overall score, we apply weighted average over the three categories (Single-Turn:Multi-Turn:Irrelevance=2:3:1) following official settings. For τ^2-Bench, the results are average@3.

Baselines. We compare against uniform random selection and five selectors that represent quality, difficulty, diversity, or a conventional combination of these signals. 1) RANDOM directly samples B of turn-level examples randomly. 2) ATLAS (Chen et al., 2025c) uses an LLM-derived critical-step signal and represents quality selection at the step level. 3) LLMEVAL scores each turn-level candidate with an LLM, following the quality-filtering idea similar to ALPAGASUS (Chen et al., 2024a); the prompt is adapted to tool-using and agent scenarios rather than ordinary instruction pairs (see Appendix A.3). 4) IFD (Li et al., 2024b) uses instruction-following difficulty computed from conditional likelihoods, and therefore provides a model-aware difficulty baseline. 5) MIG (Chen et al., 2025b) uses semantic information gain as a conventional instruction-data diversity baseline. 6) D3 (Zhang et al., 2025) combines conventional diversity, difficulty, and dependability (i.e., quality, we use the same scoring results as LLMEVAL) signals. While most of them are not originally proposed for tool-using data selection, we have done minor adaptation, including enabling multi-turn context and checking action-level appropriateness when considering quality.

Training and Evaluation Settings. For every selected subset, we fine-tune the same Qwen3 (Yang et al., 2025a) model with the same optimizer, learning rate, number of epochs, global batch size, maximum sequence length, and others (see Appendix A.2). We train Qwen3-1.7B, Qwen3-4B, and Qwen3-8B separately to evaluate whether the selection signal transfers across model scales. During training, only the selected target assistant turn contributes to the supervised loss.

For evaluation results, we report the mean accuracy over at least two independent runs with different random seeds when the selection methods induce randomness. To alleviate the inference randomness, we set the decoding temperature as 0 during evaluation for all agent models and user simulators (if needed). We use DeepSeek-V4-Flash (Xu et al., 2026a) as user simulators for τ^2-Bench and ACEBench Agent Multi-Turn category. This model is also used for LLM-assisted annotation in any sampling methods that need frontier LLMs (except for those need learner-relative evaluation).

4.2 RESULTS

4.2.1 OVERALL COMPARISONS

Table 1 compares all methods across model scales and agent benchmarks. In overall, nearly all compared baselines achieve a higher average score than RANDOM baseline, except that ATLAS performs worse on 4B and 8B settings. This indicates that focusing only on critical steps may overlook the diversity of states, tools, and intermediate behaviors needed for robust agentic performance.

[1] https://huggingface.co/datasets/nvidia/Nemotron-Post-Training-Dataset-v1

Table 1: Main comparison results (accuracy, %) on three benchmarks. The best performance in each category within each model is highlighted in **bold**, while the second best result is underlined. "Qual.", "Tot.", "Dif.", "Div" are short for "Quality", "Total", "Difficulty", and "Diversity".

Methods	BFCL				τ^2-Bench			ACEBench			Overall
	Single Turn	*Multi Turn*	*Irrel.*	*Overall*	*Retail*	*Airline*	*Avg.*	*Multi Turn*	*Multi Step*	*Avg.*	*Avg.*
Qwen3-1.7B											
BASE	**78.14**	10.75	77.11	44.27	8.77	10.67	9.72	3.3	10.0	6.7	20.23
RANDOM	77.57	12.06	81.21	45.43	14.47	12.00	13.23	3.3	10.0	6.7	21.79
ATLAS (Step Qual.)	78.04	16.12	81.66	47.68	13.16	13.00	13.08	3.3	**25.0**	**14.2**	24.98
LLMEVAL (Tot. Qual.)	76.60	16.37	**82.13**	47.41	13.74	11.33	12.54	3.3	15.0	9.1	22.61
IFD (Dif. Only)	76.54	12.88	81.68	45.57	10.96	**16.00**	13.48	6.7	15.0	10.8	23.28
MIG (Div. Only)	76.95	16.25	79.28	46.99	14.92	14.67	14.80	3.3	15.0	9.1	23.63
D3 (Joint)	77.64	14.50	75.86	45.77	14.04	10.00	12.02	**10.0**	15.0	12.5	23.43
ACE-SAMPLER	77.39	**19.38**	78.37	**48.55**	**15.79**	**16.00**	**15.90**	6.7	15.0	10.9	**25.12**
Qwen3-4B											
BASE	**86.13**	30.25	79.32	57.05	34.50	26.67	30.58	16.7	15.0	15.9	34.51
RANDOM	84.89	33.94	**82.09**	58.95	41.23	27.34	34.28	21.7	37.5	29.6	40.94
ATLAS (Step Qual.)	85.22	33.75	80.89	58.76	37.72	28.67	33.20	13.3	**45.0**	29.1	40.35
LLMEVAL (Tot. Qual.)	84.84	39.25	81.93	61.56	39.47	30.00	34.74	13.3	40.0	26.7	41.00
IFD (Dif. Only)	84.70	34.62	81.77	59.17	39.18	**31.33**	35.26	13.3	**45.0**	29.1	41.17
MIG (Div. Only)	84.24	39.12	80.41	61.04	37.43	29.33	33.38	**30.0**	35.0	32.5	42.31
D3 (Joint)	84.98	42.25	77.80	62.42	37.72	30.00	33.86	20.0	35.0	27.5	41.26
ACE-SAMPLER	85.10	**42.88**	80.28	**63.18**	**43.86**	31.30	**37.58**	23.3	**45.0**	**34.2**	**44.99**
Qwen3-8B											
BASE	83.99	36.38	78.80	59.32	43.86	30.00	36.93	20.0	35.0	27.5	41.25
RANDOM	**85.07**	43.50	**82.19**	63.80	40.94	34.67	37.80	20.0	35.0	27.5	43.03
ATLAS (Step Qual.)	84.74	43.75	82.09	63.80	41.23	33.00	37.12	23.3	25.0	24.2	41.71
LLMEVAL (Tot. Qual.)	84.85	45.88	80.81	64.69	42.98	34.00	38.49	23.3	35.0	29.1	44.09
IFD (Dif. Only)	84.59	43.62	81.81	63.64	42.55	**36.00**	39.28	20.0	30.0	25.0	42.64
MIG (Div. Only)	84.55	44.25	79.22	63.51	41.23	34.00	37.61	23.3	**45.0**	34.2	45.11
D3 (Joint)	83.34	**47.00**	79.21	64.48	45.61	32.00	38.81	**40.0**	42.5	**41.3**	48.20
ACE-SAMPLER	84.48	46.88	80.94	**65.09**	**46.49**	**36.00**	**41.25**	33.3	**45.0**	39.2	**48.51**

ACE-SAMPLER achieves the strongest aggregate performance at all three model scales. Specifically, its overall score is 25.12 for Qwen3-1.7B, 44.99 for Qwen3-4B, and 48.51 for Qwen3-8B, compared with 21.79, 40.94, and 43.03 for RANDOM, respectively. It shows that the selection signal transfers across model sizes rather than being tied to a single learner. The largest and most consistent improvements appear on interaction-heavy metrics, including BFCL Multi-Turn and τ^2-Bench. These results are consistent with the intended role of hierarchical behavioral coverage: it is most useful when success depends on choosing valid actions over multiple interaction steps, rather than only on producing a plausible isolated response.

The baselines reveal that no single conventional signal is uniformly reliable for agent data. Difficulty-only and Diversity-only selectors each improve some subsets but regress on others; for example, IFD is relatively strong on τ^2-Bench, whereas MIG is competitive on its ACEBench average and BFCL Multi-Turn. D3 is a baseline considering joint effects of several sources, yet still lay behind our method as its estimation is not agent-specific.

Another finding regards the Single-Turn and Irrelevance categories in BFCL, where all compared methods share similar performance and sometimes even lower than the models without training. Most

Table 2: Results for three accuracy gates based on Qwen3-4B.

Method	Benchmark	Score
ATLAS	BFCL-MT	33.75
LLMEVAL		39.25
Rule-based		37.38
ATLAS	τ^2-Retail	37.72
LLMEVAL		39.47
Rule-based		41.52

Table 3: Rule-based filtering statistics over the candidate pool.

Filter rule	Percentage
Invalid tool definition	6.72%
Possible argument hallucinations	2.87%
Assistant content too long	0.91%
Tool arguments violate tool definition	0.51%
No tool usage while claiming using tools	0.23%
Call to non-existent tools	0.17%
Total filter samples	**11.43%**

models have already saturated on the simple single-turn cases and most training sources are more likely to target on multi-turn scenarios (see Table 6).

4.2.2 ABLATION STUDY

Accuracy Evaluation. We first isolate how candidate validity is estimated. We compare the rule-based validity filter, an LLM score over the complete candidate responses (LLMEval), and the critical-step signal used by ATLaS. To alleviate the impact from other modules, we just randomly shuffle the candidate turn pool and then sample the instances based on the three different methods.

The results (Table 2) on BFCL Multi-Turn and τ^2-Retail show that rule-based filtering achieves competitive results to LLM-based evaluation. While more detailed and specifically designed prompting may have further potential (Barke et al., 2026), it exceeds the scope of this paper and we leverage rule-based filtering for simplicity. We further display the error type distribution figured out by our rule-based checker in Table 3. We can find that some basic rules indeed find out possible hallucinations caused by data generator (for example, we make some ID-like parameters should exsit in the context history), and therefore may influence the training performance.

Agentic Diversity. We compare two representations for diversity, semantic-level embeddings and our proposed factor-level representations, while keeping the accuracy and complexity estimators fixed. We compare in three settings, MIG (only considering diversity), D3 (also considering difficulty and dependability), and our method ACE-SAMPLER. The results are displayed in Table 4. It can be observed that when only considering diversity (MIG), embedding-level representation yields slightly better average results in two benchmarks. When we also considering accuracy and complexity, our factor-level representation contributes more to final performance. It may be due to embeddings contain more information rather than only diversity and when only considering it during sampling should be more informative. When combining with more anchors, factors look like a better choice.

Table 4: Semantic embeddings and agentic factors are compared as diversity representations.

Method	Diversity	BFCL-MT	τ^2-Retail
MIG	Embedding	39.12	37.43
	Factor	38.12	37.72
D3	Embedding	42.25	37.72
	Factor	41.62	39.18
ACE-SAMPLER	Embedding	42.25	42.40
	Factor	42.88	43.86

Complexity Estimation. For complexity, we choose to use model-aware pass@k-like metrics introduced in previous work (Zeng et al., 2026), rather than probability-based evaluation mostly used in instruction data (Li et al., 2024b; Zhang et al., 2025), as reasoning content introduces noises when calculating output probability and agentic scenarios may pay more attention on some minor but critical differences like parameter values. Here we further conduct experiments to validate this choice. We compare our complexity score, with NLL (Negative Log-Likelihood, for better comparison, is also normalized to $[0, 1]$). The baseline IFD is also compared.

Results in Figure 2 show that sampling based on medium difficulty score achieves the best performance for both difficulty scores, indicating the importance of learner-aware sampling. Our method achieves better performance when selecting the medium complexity as selector center, and outperforms the IFD, validating that it is more suitable in tool-using scenarios.

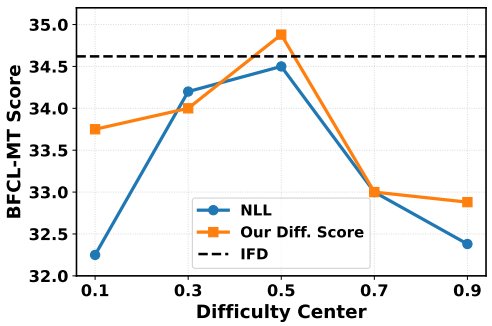

Figure 2: Difficulty estimation comparison.

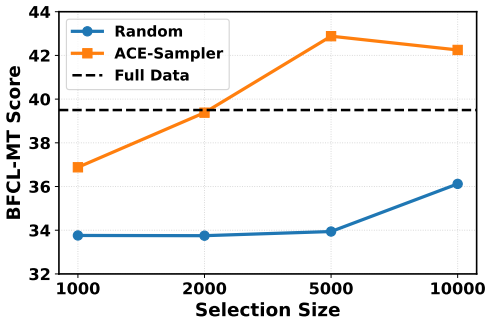

Figure 3: Comparison with different data scales.

4.2.3 Data Scaling

To study the quality–quantity trade-off, we construct subsets with increasing budgets while holding the candidate pool, selector, and training recipe. The compared budgets include 1K, 2K, 5K, and 10K. We also compared with training with full available turn-level training samples (approximately 180K). We compare ACE-Sampler with Random at each budget and plot accuracy on BFCL-MT against the number of selected turns. The results are shown in Figure 3. As can be observed, our method achieves a much faster improvement before 5K budget, and already surpasses results trained with full data. This may be because the accuracy gate hasn't been applied to the full set (there are over 10% problematic samples according to Table 3), which can hinder the performance. On the other hand, a larger budget (10K) cannot observe further improvement in ACE-Sampler, indicating the performance may have been saturated.

4.2.4 Data Analysis

This section analyzes the distribution of the final 5,000 samples selected by ACE-Sampler across model scales. Table 5 reports the source composition, the proportion of non-tool-calling turns, and the position of tool calls within a trajectory.

Table 5: Distributions of the 5,000 turn-level subsets selected for the three model scales.

Category	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
EnvFactory	1,765	1,853	1,938
EnvScaler	1,572	1,539	1,475
Nemotron	771	677	703
ToolACE	458	482	447
Toucan	329	345	338
APIGen-MT	98	94	92
ToolMind	7	10	7
Non call turn	115	120	117
First step turn	3,053	3,039	2,949

The source composition is dominated by the two environment-based datasets EnvFactory and EnvScaler. This is consistent with the selector favoring trajectories that expose richer state-dependent interaction patterns. The remaining sources share similar scales except for APIGen-MT and ToolMind. APIGen-MT focuses on τ-Bench tools and is therefore been considered as redundancy, while a large amount of ToolMind candidate tools violate the normal Jsonschema definition and is therefore filtered out by the rule checker. As the model scale increases, it is found that the selector favor more EnvFactory dataset, indicating that it may contain more samples with higher complexity.

Another interesting finding is that non-tool-call turns are selected only about 2% proportion but the trained model still perform well on multi-turn benchmarks. And over 60% selected tool-call turns are the first step of the action, indicating that the first determining actions are generally more critical.

5 Conclusion

We introduced ACE-Sampler, a hierarchical stochastic sampler that combines behavioral diversity, accuracy filtering, and learner-aware complexity for tool-use data selection. Structured behavioral factors guide trajectory-level coverage and importance-guided turn selection under a fixed training budget. ACE-Sampler outperforms baselines across three benchmarks and model scales, improving aggregate accuracy over random selection by more than 4 percentage points on average.

AI USE STATEMENT

We used generative AI tools for language editing, statement refinement, related work discovery, and coding assistance. The research questions, methodological choices, and final implementation decisions were made by the authors; generative AI was not used to originate the ideas or to directly implement the proposed method. The authors reviewed and verified all AI-assisted text and code, including their technical claims and behavior. We take responsibility for the final content of this work and for all artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the selection of tool-use training data, and the candidate pools are constructed from publicly released datasets and processed according to their applicable licenses. We do not introduce new personal data or deploy an agent in a real-world decision-making setting; nevertheless, inherited privacy, licensing, and representational biases in the source data may remain. Our rule-based validity checks reduce structural and formatting errors but do not guarantee factual correctness, safety, or suitability for every downstream application, so selected data should be audited before deployment.

REFERENCES

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. AgentRx: Diagnosing ai agent failures from execution trajectories, 2026. URL https://arxiv.org/abs/2602.02475.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment, 2025. URL https://arxiv.org/abs/2506.07982.

Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Yuefeng Huang, Xiangcheng Liu, Wang Xinzhi, and Wu Liu. ACEBench: A comprehensive evaluation of LLM tool usage. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12970–12998, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.697. URL https://aclanthology.org/2025.findings-emnlp.697/.

Guhong Chen, Chenghao Sun, Cheng Fu, Qiyao Wang, Zhihong Huang, ChaoPeng Wei, Guangxu Chen, Feiteng Fang, Ahmadreza Argha, Bing Zhao, Xander Xu, Qi Han, Hamid Alinejad-Rokny, Qiang Qu, Binhua Li, Shiwen Ni, Min Yang, HU Wei, and Yongbin Li. Beyond quantity: Trajectory diversity scaling for code agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15676–15691, San Diego, California, United States, 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.768. URL https://aclanthology.org/2026.findings-acl.768/.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=FdVXgSJhvz.

Yicheng Chen, Yining Li, Kai Hu, Ma Zerun, HaochenYe HaochenYe, and Kai Chen. MIG: Automatic data selection for instruction tuning by maximizing information gain in semantic space. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9902–9915, Vienna, Austria, 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.515. URL https://aclanthology.org/2025.findings-acl.515/.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-FLAN: Designing data and methods of effective agent tuning for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9354–9366, Bangkok, Thailand, 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.557. URL https://aclanthology.org/2024.findings-acl.557/.

Zhixun Chen, Ming Li, Yuxuan Huang, Yali Du, Meng Fang, and Tianyi Zhou. ATLAS: Agent tuning via learning critical steps. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25334–25349, Vienna, Austria, 2025c. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1299. URL https://aclanthology.org/2025.findings-acl.1299/.

Qirun Dai, Dylan Zhang, Jiaqi W. Ma, and Hao Peng. Improving influence-based instruction tuning data selection for balanced learning of diverse capabilities. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7079–7102, Suzhou, China, 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.373. URL https://aclanthology.org/2025.findings-emnlp.373/.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, Jiajie Jin, Yutao Zhu, Hanbin Wang, Fangyu Lei, Qinyu Luo, Mingyang Chen, Zehui Chen, Jiazhan Feng, Ji-Rong Wen, and Zhicheng Dou. Agent-World: Scaling Real-World Environment Synthesis for Evolving General Agent Intelligence, 2026. URL https://arxiv.org/abs/2604.18292.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. MoDS: Model-oriented Data Selection for Instruction Tuning, 2023. URL https://arxiv.org/abs/2311.15653.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14255–14273, Bangkok, Thailand, 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.769. URL https://aclanthology.org/2024.acl-long.769/.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7602–7635, Mexico City, Mexico, 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.421. URL https://aclanthology.org/2024.naacl-long.421/.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. One-shot learning as instruction data prospector for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4586–4601, Bangkok, Thailand, 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.252. URL https://aclanthology.org/2024.acl-long.252/.

Zhongzhi Li, Yucheng Shi, Zongxia Li, Ruhan Wang, Anhao Li, Zixun Huang, Junyao Yang, Lei Ke, Ninghao Liu, Haitao Mi, and Leowei Liang. Recursive Synthesis for Long-Horizon Terminal Tasks, 2026. URL https://arxiv.org/abs/2608.05466.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December*

10 - 15, 2024, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/b130a5691815f550977e331f8bec08ae-Abstract-Conference.html.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=BTKAeLqLMw.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. Toolace: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=8EB8k6DdCU.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh R. N., Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024c. URL http://papers.nips.cc/paper_files/paper/2024/hash/61cce86d180b1184949e58939c4f983d-Abstract-Datasets_and_Benchmarks_Track.html.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=pszewhybU9.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/5e3661f7fe4c8ac5652d62eb3d3c96ea-Abstract-Datasets_and_Benchmarks_Track.html.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. EnvScaler: Scaling tool-interactive environments for LLM agent via programmatic synthesis. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8326–8357, San Diego, California, United States, 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.407. URL https://aclanthology.org/2026.findings-acl.407/.

Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. Agent World Model: Infinity Synthetic Environments for Agentic Reinforcement Learning, 2026. URL https://arxiv.org/abs/2602.10090.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PG5fV50maR.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026a. URL https://arxiv.org/abs/2606.19348.

Minrui Xu, Zilin Wang, Mengyi DENG, Zhiwei Li, Zhicheng Yang, Xiao Zhu, Yinhong Liu, Boyu Zhu, Baiyu Huang, Chao Chen, Heyuan Deng, Fei Mi, Lifeng Shang, Xingshan Zeng, and Zhijiang Guo. EnvFactory: Scaling Tool-Use Agents via Executable Environments Synthesis and Robust RL, 2026b. URL https://arxiv.org/abs/2605.18703.

Zhangchen Xu, Adriana Meza Soria, Shawn Tan, Anurag Roy, Ashish Sunil Agrawal, Radha Poovendran, and Rameswar Panda. TOUCAN: Synthesizing 1.5M Tool-Agentic Data from Real-World MCP Environments, 2025. URL https://arxiv.org/abs/2510.01179.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Chen Yang, Ran Le, Yun Xing, Zhenwei An, Zongchao Chen, Wayne Xin Zhao, Yang Song, and Tao Zhang. ToolMind Technical Report: A Large-Scale, Reasoning-Enhanced Tool-Use Dataset, 2025b. URL https://arxiv.org/abs/2511.15718.

Yuming Yang, Yang Nan, Junjie Ye, Shihan Dou, Xiao Wang, Shuo Li, Huijie Lv, Tao Gui, Qi Zhang, and Xuanjing Huang. Measuring data diversity for instruction tuning: A systematic analysis and a reliable metric. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18530–18549, Vienna, Austria, 2025c. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.908. URL https://aclanthology.org/2025.acl-long.908/.

Xingshan Zeng, Weiwen Liu, Xu Huang, Zezhong Wang, Lingzhi Wang, Liangyou Li, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruiming Tang, and Qun Liu. Toolace-r: Model-aware iterative training and adaptive refinement for tool learning. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 34593–34601. AAAI Press, 2026. doi: 10.1609/AAAI.V40I41.40759. URL https://doi.org/10.1609/aaai.v40i41.40759.

Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. D3: diversity, difficulty, and dependability-aware data selection for sample-efficient LLM instruction tuning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 8348–8356. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/928. URL https://doi.org/10.24963/ijcai.2025/928.

Yang Zhao, Li Du, Xiao Ding, Yangou Ouyang, Hepeng Wang, Kai Xiong, Jinglong Gao, Zhouhao Sun, Dongliang Xu, Qing Yang, Dongchen Li, Bing Qin, and Ting Liu. Beyond similarity: A

gradient-based graph method for instruction tuning data selection. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24391–24404, Vienna, Austria, 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1189. URL https://aclanthology.org/2025.acl-long.1189/.

A Appendix

A.1 Dataset Statistics

We construct the candidate pool from seven publicly available sources. Table 6 reports the number of trajectories retained after the common preprocessing step, the average number of user turns and tool-calling assistant turns per trajectory, and whether the source contains explicit reasoning annotations. We define a *user turn* as one user message and a *tool turn* as an assistant message containing at least one tool call. A multi-step step is measured between two consecutive user messages and includes the intervening tool calls. Dataset sizes and averages are descriptive statistics of the processed candidate pool, not claims about the original releases. Small differences can therefore arise from malformed, over-length, or otherwise invalid records removed by the common preprocessing pipeline.

Table 6: Statistics of the seven data sources used to construct the candidate pool.

Dataset	**Domain**	**Size**	**Avg. user turns**	**Avg. tool turns**	**Reasoning**
ToolACE	General Synthetic Tools	10k	1.11	0.94	No
APIGen-MT	τ^2 Retail/Airline	5k	4.78	4.28	No
Toucan	General MCP Tools	120k	1.72	1.45	No
Nemotron	Mixed Domains	310k	2.28	2.30	Yes
ToolMind	Graph Synthetic	160k	2.69	2.32	Yes
EnvScaler	Executable Environments	9k	5.76	13.33	Yes
EnvFactory	Executable Environments	3k	8.14	15.09	Yes

A.2 Implementation Details

Here we show the final used hyper-parameters for our experiments. Table 7 separates the candidate search space from the final configuration. Each baseline is trained using the same training recipe.

Table 7: Candidate hyperparameter configurations and final settings for sampling and model training.

Hyperparameter	**Candidates / options**	**Final setting**
Sample budget B	1K, 2K, 5K, 10K	5K
Maximum turns per trajectory K	2, 3	3
Similarity exponent γ in Eq. 6 and 7	0.2, 0.5, 1.0, 1.5	1.5 (B =1K), 1.0 (2K), 0.5 (5K), 0.2 (10K)
Continuation factor η in Eq. 8	5	5
Difficulty center c in Eq. 9	0.1, 0.3, 0.5, 0.7, 0.9	0.5
Learning rate	1e-6, 1.5e-6, 2e-6	1e-6 (Qwen3-8B), 1.5e-6 (1.7B/4B)
Learning-rate scheduler	Cosine	Cosine
Number of epochs	3, 4, 5	5
Global batch size	16, 32, 64	32
Maximum sequence length	16K	16K (already cover most samples)

A.3 Prompts

We first show the prompt for factor-level annotation (Section 3.2). We annotate each complete trajectory in one offline LLM call. The annotator sees the ordered messages and tool definitions,

assigns trajectory-level scenario and task labels, and then labels every assistant turn with its function, interaction relation, and counterfactual importance rank. The full prompt, including the closed label vocabularies and output schema, is included below.

Trajectory Factor-level Annotation Prompt

System Prompt

You are an expert annotator of multi-turn tool-using agent trajectories. Analyze one complete trajectory and annotate every assistant turn in one response. Use only the supplied tool definitions, messages, tool observations, and causal relationships. Return exactly one valid JSON object matching the output schema; do not return Markdown, explanations, comments, or text outside the JSON object.

General Rules

1. Every input message has a unique `turn_id` (for example `user_1`, `assistant_1`, or `tool_1`).
2. Produce exactly one entry for every input assistant turn. Do not omit, duplicate, rename, or invent assistant turn IDs.
3. `depends_on_turn_ids` may contain only IDs of earlier input messages.
4. Use only the declared closed-set labels; use other or unknown when needed.
5. Judge turns from the complete trajectory, not keywords alone. Consider the goal, tool semantics, observations, and later use of each result.
6. A new independent user request starts a new task. Clarifications, confirmations, corrections, and follow-ups for the same goal do not.
7. Every assistant turn belongs to exactly one task.
8. Analyze calls by semantic purpose; a call is not automatically a core action.
9. Importance measures contribution to task completion, not verbosity, position, or writing style.
10. Do not expose or reproduce hidden reasoning. Output labels only.

Trajectory-level Labels

For every task, assign one `primary_scenario` and zero to two `secondary_scenarios`. Use secondary scenarios only when they materially change required behavior. The closed scenario set is: `finance`, `office_productivity`, `daily_life`, `travel`, `entertainment`, `education`, `devices_iot`, `other`, `transportation`, `artificial_intelligence`, `business_management`, `food_dining`, `biology_nature`, `communications`, `security`, `home`, `technology`, `weather_climate`, `geography`, `society_politics`, `retail_commerce`, `healthcare`, `software_data`, `legal_government`.

For every task, assign one `primary_task_type` and zero to two secondary types. Label the end goal rather than every prerequisite. Secondary types must be material subgoals. The closed task-type set is: `retrieve`, `create`, `update`, `delete_or_cancel`, `transact_or_book`, communicate, `compute_or_transform`, `analyze_or_compare`, `verify_or_monitor`, `troubleshoot`, `identity_or_access`, `plan_or_recommend`, `other`.

Turn-level Labels

Assign exactly one `turn_function` to every assistant turn: `acquire_information`, `request_missing_information`, `request_clarification`, `request_confirmation`, `reason_or_plan`, `perform_core_action`, `verify_result`, `retry_or_recover`, `deliver_intermediate_result`, `deliver_final_result`, refuse, handoff, `social_or_polite`, `other`.

Assign exactly one `interaction_relation` and list the specific earlier messages on which it depends: `independent`, `uses_previous_tool_result`, `uses_earlier_tool_result`, `uses_user_followup`, `retry_after_failure`, `alternative_after_failure`, `verify_previous_action`, `deliver_tool_result`, `blocked_needs_user_input`, `no_semantic_progress`, `unknown`. Use the dominant relation. Use the minimal earlier IDs needed to justify it. Use an empty dependency list for `independent`, `no_semantic_progress`, or `unknown`.

Importance

Trajectory Factor-level Annotation Prompt (continued)

Assess each turn counterfactually: if it were removed or incorrect, how much would task completion be affected? Use `critical` only for an irreplaceable step whose removal normally fails the task or causes a materially wrong outcome. Use `important` for substantial progress that could still be repaired; `supporting` for useful but non-central context; and `incidental` for courtesy or redundancy. Assign one `importance_tier`: `critical`, `important`, `supporting`, or `incidental`. Assign a dense task-local `importance_rank` (1 is most important). Ties are allowed only for genuinely equivalent impact. Rank every assistant turn.

Output Schema

Return an object with exactly these top-level fields:

```
{
  "tasks": [{
    "task_id": "task_1",
    "assistant_turn_ids": ["assistant_1", "assistant_2"],
    "primary_scenario": "travel",
    "secondary_scenarios": ["finance"],
    "primary_task_type": "transact_or_book",
    "secondary_task_types": ["retrieve"]
  },
  ...
  ],
  "assistant_turns": [{
    "turn_id": "assistant_1",
    "task_id": "task_1",
    "turn_function": "acquire_information",
    "interaction_relation": "independent",
    "depends_on_turn_ids": [],
    "importance_tier": "critical",
    "importance_rank": 1
  },
  ...
  ]
}
```

Before returning, silently verify that every assistant ID occurs exactly once, every task reference exists, every dependency points to an earlier message, all labels belong to their closed sets.

User Prompt Template

Annotate the following complete trajectory. All assistant turns must be annotated in this one response.

```
<trajectory>
{TRAJECTORY_DATA}
</trajectory>
```

Below is the prompt for LLM-based quality evaluation, using for baseline LLMEVAL and D3 (the corresponding dependability evaluation).

Tool-Using Data Quality Evaluation Prompt

Role and Evaluation Scope

You are an expert evaluator of tool-using data. Assess the quality of [Response] as the next assistant turn given [History]. Assign one overall integer score from 1 to 10.

Tool-Using Data Quality Evaluation Prompt (continued)

Evaluate whether this turn provides a correct, grounded, and useful example of agent behavior at the current state. Use the supplied instructions, tool definitions, prior messages, and observations.

Evaluation Criteria

1. Task Alignment and Progress
Does the response address the active user goal, respect constraints, and take an appropriate next step? A necessary clarification, justified refusal, or accurate progress report can be as appropriate as a tool call.

2. Tool Selection and Action Validity
When tools are used, are they appropriate and available under the supplied definitions? Are calls syntactically valid and arguments consistent with required fields, types, allowed values, and tool semantics?

3. Context, State, and Dependency Consistency
Are current choices or states consistent with the history? Are prerequisites satisfied and dependencies respected? Parallel calls must not rely on each other's unavailable results.

4. Grounding and Result Interpretation
Are claims and argument values supported by the user input, tool observations, or valid derivations? Does the response interpret outputs correctly, and avoid invented tool results or unsupported success claims?

5. Interaction and Recovery
Does the response appropriately handle missing information, ambiguity, tool errors, and user corrections when they occur? Clarifications should target genuine blockers; retries or alternative actions should address the observed failure rather than repeat it blindly.

6. Fluency and Clarity
Is user-facing text coherent, relevant, and understandable? Are tool calls and structured outputs clear and compliant with the required format?

Scoring Rules

Apply only relevant criteria. Judge this response, not the overall difficulty of the task or errors made earlier in the history. Do not require a single intermediate turn to finish the entire task. Do not reward length, tool-call count, complexity, or stylistic polish by themselves. A short, simple, correct turn can score 10. Correctness, grounding, and authorization take priority over presentation.

Input

```
[History]
{history}

[Response]
{response}
```

Output Requirement

Return exactly one integer from 1 to 10. Do not include a label, explanation, Markdown, JSON, or any other text.

A.4 Sampling Algorithm

We display the algorithm for the whole sampling procedure of our method in Algorithm 1.

Algorithm 1 Hierarchical Stochastic Sampling

Require: Trajectory pool $\mathcal{D}$, target budget B, maximum turns per trajectory K, learner $\mathcal{M}$
Ensure: Selected turn-level training set $\mathcal{S}$

Initialization: $\mathcal{S} \leftarrow \emptyset$, selected trajectories $\mathcal{R} \leftarrow \emptyset$, and shuffle $\mathcal{D}$
while $|\mathcal{S}| < B$ and unvisited trajectories remain **do**
  Sample the next trajectory τ from $\mathcal{D}$
  $p_\tau \leftarrow (1 - s_\tau)^\gamma$, where $s_\tau \leftarrow \max_{\tau' \in \mathcal{R}} S_{\text{traj}}(\tau, \tau')$ ($s_\tau = 0$ if $\mathcal{R} = \emptyset$)
  if $\text{Bernoulli}(p_\tau) = 0$ **then**
    continue *// Select the trajectory τ with probability of p_τ*
  end if
  $\pi_\tau \leftarrow \text{Sort}_{\text{imp}}(\mathcal{A}_\tau)$ *// Rank the candidate assistant turns $\mathcal{A}_\tau$ by turn importance*
  $k \leftarrow 0$
  for $a_t \in \pi_\tau$ **do**
    if a_t is invalid **then**
      continue *// Apply the validity check in turn-level*
    end if
    $p_t^{\text{div}} \leftarrow (1 - s_t)^\gamma$, where $s_t \leftarrow \max_{a' \in \mathcal{S}} S_{\text{turn}}(a_t, a')$ ($s_t = 0$ if $\mathcal{S} = \emptyset$)
    $p_t^{\text{dif}} \leftarrow 1 - \left(2|s_t^{\text{dif}} - 0.5|\right)^q$, where s_t^{dif} is obtained based on $\mathcal{M}$ *// Model-aware difficulty*
    $p_t \leftarrow p_t^{\text{div}} \cdot p_t^{\text{dif}}$
    if $\text{Bernoulli}(p_t) = 1$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{d(\tau, a_t)\}$ *// Select the turn a_t with probability of p_t*
      $k \leftarrow k + 1$
      if $k = K$ or $|\mathcal{S}| = B$ **then**
        break
      end if
      $p_{\text{cont}} \leftarrow \dfrac{|\mathcal{A}_\tau|}{|\mathcal{A}_\tau| + \eta} \dfrac{1}{2^k}$ *// Determine whether to select more turns*
      $k \leftarrow k + 1$
      if $\text{Bernoulli}(p_{\text{cont}}) = 0$ **then**
        break
      end if
    end if
  end for
  if $k > 0$ **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{\tau\}$ *// Add the trajectory into selected set*
  end if
end while
return $\mathcal{S}$ *// Final selected training turns*