ANCHOR-AND-FRONTIER: SCHEDULING TOOL-SCHEMA EXPOSURE ALONG AGENT TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using language models repeatedly ingest executable schemas while acting, creating a trade-off between full-catalog exposure and compact-context efficiency. In multi-step trajectories, however, an omitted tool can alter subsequent calls, observations, and decisions, making schema exposure a trajectory-level scheduling problem rather than only a per-turn retrieval problem. We introduce Anchor-and-Frontier (A&F), a deterministic controller that restores broad schema context at observable user-turn and execution-error checkpoints and otherwise exposes a cardinality-capped, state-conditioned frontier informed by a frozen demonstration-derived next-tool prior, without additional generative-model calls. On the BFCL V4 primary split, an always-compact controller reduces model-token use by 19.66% but loses 21.57 task-success percentage points relative to full-schema exposure, motivating selective broad-context restoration. On a predeclared 394-example condition holdout, Qwen3.5-27B with A&F achieves 259/394 successes versus 258/394 for full schema while reducing model tokens by 15.85%; a later same-condition Qwen3.5-122B analysis shows a similar 15.55% reduction. Across additional analyses and transfer experiments, token savings are more consistent than success retention, indicating that the efficiency–reliability trade-off is setting-dependent. The evidence supports the combined scheduling-and-routing system under held-out interaction conditions, rather than isolating the contribution of anchor placement.

1 INTRODUCTION

Language-model agents interleave reasoning, function calls, execution feedback, and conversational responses (Yao et al., 2023; Schick et al., 2023; Qin et al., 2024). At each decision, the model needs the names and argument structures of relevant tools. Repeatedly providing a complete catalog avoids schema omission but consumes prompt capacity that could otherwise hold the evolving interaction. This motivates compact tool retrieval, particularly as catalogs and trajectories grow (Patil et al., 2023; Hao et al., 2023).

Retrieval quality alone does not characterize a multi-step exposure policy. An early omission may change the chosen call, the returned observation, and the history used for subsequent retrieval. Conversely, repeatedly displaying the complete catalog after execution enters a well-defined local state can be redundant. Dynamic retrieval, iterative query planning, and adaptive shortlist depth address important parts of this problem (Patel et al., 2026; Fang & Glass, 2026; Repantis et al., 2026). We study a complementary question: *when should an agent restore broad schema context during an executed trajectory?*

Our hypothesis is that observable interaction events provide useful restoration points. A new user turn can introduce a new branch; an execution error signals that the current local plan may need revision. Between these events, a compact frontier can be constructed from the request, recent calls and results, schema-derived dependencies, and a frozen next-tool prior. A&F implements this hypothesis without a separate planner, verifier, or confidence-estimation call.

The main result is an end-to-end comparison against full-schema exposure on held-out multi-turn conditions from the Berkeley Function Calling Leaderboard (BFCL). With Qwen3.5-27B, A&F obtains 259/394 successes versus 258/394 for full schema and saves 15.85% of model tokens. A later exhaustive 122B analysis shows a similar operating point. Separately, on the BFCL primary split, an

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.

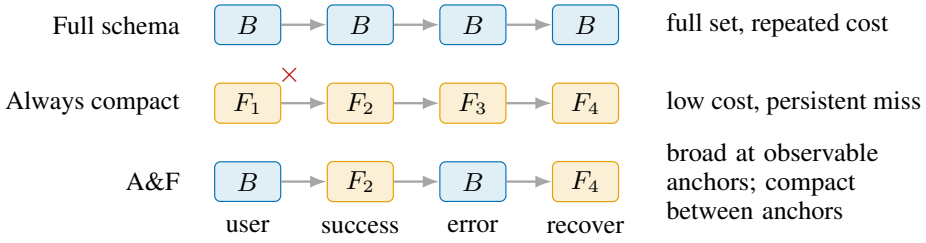

Figure 1: Illustrative exposure schedule, not a measured comparison. B denotes broad context and F_t a cardinality-capped frontier. A&F restores broad context at observable checkpoints and uses compact frontiers between them; broad exposure does not guarantee task success.

earlier always-compact state graph saves 19.66% of tokens but loses 21.57 success points. These are different splits and controller configurations: the primary deficit motivates the study but is not a same-router ablation of the confirmed schedule.

Our contributions are: (i) an observable-event controller separating broad restoration from compact frontier construction; (ii) a predeclared, paired BFCL evaluation of token use and aggregate success retention; and (iii) coverage, scenario-overlap, and transfer analyses delimiting this operating point.

Scope. The BFCL confirmation holds out conditions, not all scenario identities: the prior and schedule selection use overlapping scenarios. Scenario-disjoint sensitivities and ToolSandbox transfer do not confirm the same margin. Both model scales belong to Qwen3.5, and alternative anchor placements are not budget-matched. Thus the evaluated treatment is the complete scheduling-and-routing system.

2 Related Work

Learning and prompting tool use. ReAct interleaves reasoning with actions, and Toolformer learns tool-use behavior from self-supervised data (Yao et al., 2023; Schick et al., 2023). Gorilla combines retrieval-aware training with API collections, ToolLLM learns from thousands of APIs, and ToolkenGPT represents tools through learned embeddings (Patil et al., 2023; Qin et al., 2024; Hao et al., 2023). Our focus is inference-time exposure of original executable schemas, rather than learning tool embeddings or modifying the executing model.

Dynamic retrieval and tool-context memory. DTDR conditions retrieval on an evolving call plan, while ToolQP issues iterative sub-goal queries (Patel et al., 2026; Fang & Glass, 2026). H-EPM combines procedural transitions with episodic state, SGA-MCTS retrieves state–goal–action experiences, and HYSET predicts complementary tool sets (Li et al., 2026; Xie et al., 2026; Hong et al., 2026). MemTool manages tool contexts across multi-turn conversations using autonomous, deterministic-workflow, and hybrid modes (Lumer et al., 2025). Within this line of work, A&F separates *when* to restore broad context from *which* schemas to retain between restorations: observable user and execution-error checkpoints trigger the former, while a frozen state-conditioned router handles the latter under a fixed compact cap.

Adaptive exposure and schema efficiency. Instruction–Tool Retrieval (ITR) narrows both instructions and tools with confidence-gated fallback (Franko, 2025). Adaptive shortlist policies vary the number of tools exposed, while schema compression reduces the representation cost of selected tools (Repantis et al., 2026; Sakizli, 2026). A&F uses observable events rather than confidence to trigger restoration. All paired BFCL rows retain the original executable serialization. Our experiments do not establish superiority over ITR or MemTool, nor test composition with schema compression; these are related approaches rather than implemented baselines.

Evaluation. StableToolBench targets stable tool-use evaluation, ToolSandbox provides stateful conversation and dynamic-milestone scoring, and BFCL supplies official multi-turn execution (Guo

Table 1: Frozen BFCL frontier construction. Limits are candidate proposals, not reserved slots. The final union contains at most $K = 16$ available schemas.

Candidate source	Proposal limit	Information supplied
Lexical order	8	Request/schema term match
Structured cross-encoder	8	Semantic tool compatibility
Dependency successors	8; at most 4/source	Output-to-input handoffs
Latest sub-goal	4	Within-turn request refinement
Frozen next-tool prior	4	Observed-prefix continuation
Fusion and backfill	Final $K = 16$	Deduplicated, capped union

et al., 2024; Lu et al., 2025; Patil et al., 2025). TPS-Bench and ETOM extend evaluation toward compounding tool tasks (Xu et al., 2026; Dong et al., 2026). We distinguish official end-to-end results from controlled routing tests and local diagnostics, and retain negative outcomes.

3 METHOD

3.1 PROBLEM FORMULATION

Let $\mathcal{T}_t$ be the tool catalog available before decision t and h_t the observable interaction history. The controller selects $E_t \subseteq \mathcal{T}_t$, the set of original executable schemas exposed to the model. The evolving catalog notation accommodates tools announced during an interaction. Task success is determined by the benchmark's executed calls and final state, rather than by retrieval accuracy alone.

Let $c(E_t)$ denote the tokenizer-measured cost of the selected schemas. The objective is to reduce repeated schema exposure while retaining end-to-end task success. A fixed compact policy enforces $|E_t| \leq K$ at every decision. A&F instead selects between a broad set B_t and a compact frontier F_t. In the main BFCL experiments, $B_t = \mathcal{T}_t$; the bounded-broad large-catalog pilot is a separate variant (Appendix G).

3.2 CARDINALITY-CAPPED, STATE-CONDITIONED FRONTIER

Each external tool-memory record contains the tool name, operation type, domain, required parameters, aliases, and discriminative schema cues. The frozen router combines lexical order, a structured cross-encoder transferred from the pre-BFCL StableToolBench diagnostic pipeline, schema-derived successors of recent calls, the initial request, the latest user sub-goal, and observable successful or failed tool events. Table 1 gives its proposal limits. Deterministic deduplication, fusion, and backfill select at most $K = 16$ tools from the available catalog. These limits were selected on BFCL primary development shards 7–10 and frozen before the condition-holdout outcomes were opened.

The cap constrains *schema count*, not an exact token budget: schemas have unequal serialized lengths, and smaller catalogs may yield fewer than K entries. Compact memory records are used only for routing. Every selected tool is exposed through its unchanged full executable definition.

The model also receives a bounded state ledger of recent calls, observable results, public identifiers, failed calls, and the current frontier. Neither the ledger nor the event controller accesses evaluator milestones, gold trajectories, or hidden database state during execution. The next-tool prior is a TF–IDF unigram/bigram nearest-decision model with a categorical fallback, distilled from successful full-schema traces for 76 development scenario identities. It proposes up to four candidates and changes priority without a generative-model call. This is demonstration-derived information, not a training-free routing component. Its provenance matters because some condition-holdout identities overlap development. ToolSandbox transfer retains its parent router's $K = 2$ cap.

3.3 OBSERVABLE-EVENT EXPOSURE SCHEDULE

Let $u_t = 1$ for the first decision after an ordinary user turn. Let $e_t = 1$ while an observable execution-error checkpoint is open. An execution error opens this checkpoint; a subsequent successful tool call

Algorithm 1 A&F exposure at decision t (policy-level specification)

Require: Observable history h_t, available catalog $\mathcal{T}_t$, frozen router R, cap K
$u_t \leftarrow \mathbf{1}\{\text{first decision after an ordinary user turn}\}$
$e_t \leftarrow \mathbf{1}\{\text{execution-error checkpoint remains open}\}$
if $u_t \vee e_t$ **then**
    $E_t \leftarrow \mathcal{T}_t$ ▷ Main BFCL broad branch
else
    $E_t \leftarrow R(h_t, \mathcal{T}_t; K)$ ▷ $|E_t| \leq K$
end if
Invoke the executing model with the original schemas in E_t and its state ledger.
Record calls and visible executor outcomes for the next decision.

Table 2: Evaluation populations and their evidential roles. "Conditions" does not imply unseen scenario identities.

Evaluation	Population	Role and separation
Controlled routing	3,000 synthetic tasks	Frozen generator; oracle step decomposition
BFCL primary	394 base/long-context	Compact-routing diagnosis and development
BFCL condition, 27B	394 variants; 197 clusters	Predeclared confirmation; identity overlap
BFCL condition, 122B	Same 394 variants	Later exhaustive analysis; 120 variants seen earlier
ToolSandbox	81 scenarios $\times$ 3 trials	Frozen original matrix; A&F transfer is post-hoc
Catalog pilot	4 examples; 96 trajectories	Exploratory; repeated tasks and changed serving

closes it. Define $a_t = u_t \vee e_t$. The exposed set is

$$E_t = \begin{cases} B_t, & a_t = 1, \\ F_t, & a_t = 0. \end{cases} \tag{1}$$

A user checkpoint applies to its first subsequent decision, whereas the error checkpoint persists until a successful call is observed. Here "unresolved" is an operational status, not a semantic verification that the task-level error remains unsolved. Conversely, a successful call resets the checkpoint but need not repair every consequence of an earlier mistake.

Algorithm 1 specifies exposure, not the frozen router's implementation. Selection finishes before the executing-model call and adds no planner or verifier retry. Resetting the error checkpoint after success prevents one historical error from forcing broad exposure indefinitely.

For a *fixed realized trajectory*, the schema-token difference relative to exposing its broad set at every decision is

$$\sum_t \big[c(B_t) - c(E_t)\big] = \sum_t (1 - a_t)\big[c(B_t) - c(F_t)\big]. \tag{2}$$

This accounting identity explains where schema savings can arise; it is not the end-to-end difference between independently executed policies. Different exposure policies can change histories, calls, completions, and trajectory length. We therefore measure all model tokens on complete executed tasks. "No additional calls" means no additional *generative-model* calls; local routing still incurs computation.

4 Experimental protocol

Evidence roles. Table 2 separates controlled tests, development, predeclared confirmation, and external-validity analyses. Reference-history replay and opened-development examples are not independent end-to-end confirmations.

BFCL V4. We evaluate multi-turn subsets, not the full leaderboard. The primary split contains 394 base and long-context examples. The separately sealed condition split contains 394 missing-function and missing-parameter variants in 197 numeric scenario clusters. These conditions change when functions or parameters become available but reuse scenario identities, tool classes, and benchmark

paths. The paired Full/A&F runs use the official stateful executor, native answer path, one evaluator pass, a 32k context limit, a requested 4,096-token output cap, identical model revisions, and matched decoding. The pinned vLLM server uniformly floors requested temperature 0.001 to 0.01. The evaluated BFCL multi-turn pipeline has no LLM user simulator or model-based evaluator; its token totals include the model-dependent routing and execution-decision calls.

Selection and freezing. Candidate policies were fit on primary shards 0–6. On shards 7–10, three jointly declared ledger modes scored 53/96, 52/96, and 51/96; the success-first rule selected the minimal ledger. Development inspection identified a persistent historical-error checkpoint and a mismatch between advertised and pinned context length. Before confirmation, version 2 closed the error checkpoint after success and applied the same 32k request budget to both methods. The 27B method, prompts, margin, bootstrap seed, and expected identities were hash-locked before condition outcomes were opened. The unchanged method was later run with Qwen3.5-122B-A10B-FP8 on all 394 conditions, after a 120-example stratum had already been inspected. That analysis increases coverage but is not a second independent confirmation.

ToolSandbox and controlled tests. After a ten-example calibration prefix, 81 credential-free ToolSandbox scenarios form the frozen test. Five declared methods run three trials with Qwen3.5-9B, temperature zero, a 2,048-token output cap, the official stateful executor, and dynamic-milestone evaluation. A&F later transfers the BFCL-selected event schedule without ToolSandbox-specific label, threshold, or prompt tuning; because the original matrix had been inspected, this is post-hoc zero-shot transfer. The synthetic test uses 3,000 examples from seeds 142–144 over 10,000 tools, including 2,250 multi-step tasks. Its deterministic executor and oracle decomposition test routing and state carry, not autonomous agent competence.

Metrics and statistical inference. Task success is the official end-to-end outcome. Let T_i denote the sum of input and output tokens over model calls for task i. Token columns report means unless labeled totals; savings use the aggregate paired ratio

$$S = 1 - \frac{\sum_i T_i^{\text{candidate}}}{\sum_i T_i^{\text{full}}}. \tag{3}$$

Confidence intervals use 20,000 paired percentile-bootstrap replicates. BFCL resamples numeric scenario clusters; ToolSandbox resamples scenarios while retaining their three trials. Each replicate recomputes the aggregate token ratio; reported percentages use aggregate counts rather than rounded table means. Non-inferiority requires the lower 95% confidence bound of candidate-minus-full success to exceed -5 percentage points. The protocol also prespecifies a minimum 10% token reduction. The five-point margin is an explicit tolerance, not an equivalence or per-task preservation claim. BFCL wall-clock time, accelerator cost, and router FLOPs were not logged; local-router computation is not included in the reported token totals.

Prior-dependence replay. A post-hoc CPU audit replays the same frozen 27B router with and without the next-tool prior, inserting reference calls as successful observable history. Both variants see the same optimistic prefix. This isolates candidate inclusion at compact decisions but cannot estimate prior-free on-policy success. Additional protocol details and artifact boundaries appear in Appendix E.

5 Results

5.1 Condition-held-out BFCL success and token use

Table 3 presents the principal result. The predeclared 27B comparison obtains 259/394 successes with A&F versus 258/394 with Full. The success difference is $+0.25$ points, with paired clustered 95% CI $[-3.81, +4.31]$, satisfying the -5-point margin. Mean model tokens fall from 88,679 to 74,622: 15.85% savings with CI [13.74,17.65]%. Both the estimate and its lower confidence bound exceed the prespecified 10% token threshold.

At 27B, missing-function success is 142/197 versus 141/197 for Full; missing-parameter ties at 117/197, although its category-specific interval does not establish non-inferiority. The policies

Table 3: Official BFCL V4 on all 394 secondary-condition examples. Δ is A&F minus Full task success, in percentage points. Intervals are paired and scenario-clustered. The 27B row is predeclared; 122B-A10B-FP8 is a later same-condition analysis.

Qwen3.5	Full	A&F	Δ [95% CI]	Token saving [95% CI]
27B	258/394	259/394	$+0.25$ $[-3.81, +4.31]$	15.85% [13.74,17.65]
122B-A10B	242/394	245/394	$+0.76$ $[-3.05, +4.57]$	15.55% [13.49,17.64]

Table 4: Condition results by development overlap. The 27B strata partition all 394 examples. The 122B scenario-disjoint stratum was inspected before its exhaustive analysis. Intervals describe success differences in percentage points.

Model	Scenario stratum	Full	A&F	95% CI	Saving
27B	Policy fitting, 0–6	122/178	128/178	$[-2.81, +9.55]$	14.65%
27B	Selection, 7–10	57/96	59/96	$[-5.21, +9.38]$	17.76%
27B	Disjoint, 11–15	79/120	72/120	$[-12.50, +0.83]$	15.95%
122B	Disjoint, 11–15	78/120	78/120	$[-8.33, +8.33]$	17.74%

disagree on 59 tasks: A&F wins 30 and Full wins 29, with 335 ties. A&F uses fewer model tokens on 358/394 tasks. Thus aggregate retention coexists with task-level regressions: 29 of Full's 258 successful tasks fail under A&F. The one-success net difference is not evidence of overall superiority.

The 122B analysis obtains 245/394 successes versus 242/394, while mean tokens fall from 89,101 to 75,242. Missing-function success is 138/197 versus 136/197; missing-parameter is 107/197 versus 106/197. Both category-specific intervals satisfy the same five-point margin. This extends evaluation to a second Qwen3.5 configuration; it is not a controlled parameter-scaling study, model-family transfer, or independent confirmation.

A retrospective $K = 12$ run also scores 259/394 and uses 68,767 mean tokens, a 22.45% reduction (Appendix B). Because its outcomes were observed after the confirmation was opened, it does not replace the predeclared $K = 16$ headline.

5.2 Scenario overlap and prior dependence

The 27B condition results partition by scenario overlap in Table 4. On the 120 variants outside policy fitting and schedule selection, A&F scores 72/120 against Full's 79/120. The -5.83-point difference has CI $[-12.50, +0.83]$ and fails the non-inferiority criterion, despite 15.95% fewer tokens. The corresponding 122B stratum ties at 78/120 and saves 17.74%, but its CI $[-8.33, +8.33]$ also fails the criterion. A favorable or tied point estimate does not remove the uncertainty.

These differences do not prove memorization: difficulty may vary across shards. They do, however, limit the pooled result to condition generalization and prevent attributing it solely to the event schedule. Prior-removal replay further shows that compact-decision gold-tool recall falls from 96.23% to 85.53% on the condition set, and from 97.10% to 82.61% on primary scenario-disjoint shards 11–15. Complete-example coverage falls as well, with clustered difference intervals $[-29.25, -16.04]$ and $[-46.81, -17.02]$ points (Appendix D). The prior materially affects candidate coverage. Because replay uses reference rather than generated history, this finding cannot substitute for an end-to-end schedule-by-prior ablation.

5.3 What always-compact exposure loses

On the separate primary split, Full succeeds on 259/394 tasks and the earlier always-compact state graph on 174/394 (Table 5). The compact-minus-full difference is -21.57 points, CI $[-27.92, -15.48]$, despite 19.66% fewer model tokens, CI [16.19,23.06]%. TF–IDF, BGE-M3 (Chen et al., 2024), and the basic compact router perform worse at the same $K = 16$ cap. These standard controls were frozen after Full/state-graph outcomes were opened, so they are retrospective

Table 5: Official Qwen3.5-27B BFCL primary split, not the condition holdout. Tokens are means per task. Appendix E maps compact-router labels to archived experiment names.

Exposure/router	Success	Mean model tokens	Saving
Full schema	259/394	75,877	—
TF–IDF RAG ($K = 16$)	116/394	50,137	33.92%
BGE-M3 RAG ($K = 16$)	88/394	45,103	40.56%
Basic compact router ($K = 16$)	130/394	47,982	36.76%
Compact router + state graph ($K \leq 16$)	174/394	60,961	19.66%
smolagents (full catalog)	104/394	44,040	41.96%

Table 6: BFCL diagnostics. Top: selected primary evaluator labels. Bottom: 216-example policy confirmation on shards disjoint from policy fitting. The two panels have different populations and purposes.

Primary error label	Full schema	Always compact
Empty turn	3	18
Execution-response mismatch	31	51
Final instance-state mismatch	56	106
Long-context inference error	43	43
Policy confirmation (216 examples)	**Success**	**Mean model tokens**
Full schema	135/216	76,496
Untrained compact state graph	86/216	60,555
12-example sequence prior	83/216	59,654
Development-distilled prior	108/216	61,001

comparisons. The smolagents row[1] retains its own agent prompt and 25 parsing failures; it is an external full-catalog agent, not a same-prompt compact control.

Local failure diagnosis, not a causal partition. On predesignated development shards 0–6, all 31 compact failures with an identifiable failed turn were audited. Sixteen omit a required schema throughout that turn; 15 expose every required schema within it and still fail. Inclusion within a turn neither establishes availability at the exact decision needed nor excludes earlier omissions. These are local diagnoses, not a causal partition or a population-wide omission-rate estimate.

The aggregate error labels are consistent with multiple downstream failure modes (Table 6). Compact exposure adds 15 empty turns, 20 execution-response mismatches, and 50 final-state mismatches. Both methods have 43 long-context inference errors. Equal counts show no aggregate increase in that label; they do not establish identical affected examples or exclude other context-related mechanisms.

A development-distilled next-tool policy improves compact success from 86 to 108/216, or 10.19 points, but remains 12.50 points below Full. In its policy audit, exactly localized candidate-missing failures fall from 15 to 5, while 27 of 32 remaining auditable failures expose every required schema within the diagnosed turn. This is a separate audit from the 31-failure development sample. It supports evaluating execution in addition to candidate coverage, without establishing that later failures are causally independent of retrieval.

Appendix C retains an opened-development trajectory illustrating broad user/error checkpoints, and Appendix A reports the controlled state-carry test. Neither supplies a budget-matched schedule ablation. In the synthetic test, the structured stateful router solves all 2,250 multi-step examples (Wilson 95% interval [.9983,1.0000]) and uses 93.2% fewer four-step schema-input tokens than schema RAG; oracle decomposition limits that result to the controlled mechanism.

[1]Framework documentation: `https://huggingface.co/docs/smolagents/index`.

Table 7: ToolSandbox: 81 scenarios, three trials each. Official denotes the dynamic-milestone score. Tokens are means per trial; end-to-end (E2E) includes the user simulator. A&F is a post-hoc transfer.

Method	Success	Official	Agent tokens	E2E tokens
Full schema	136/243	.728	5,297	7,379
TF–IDF RAG	87/243	.572	3,651	6,313
Basic compact router	100/243	.607	3,136	5,382
Compact router + state graph	98/243	.583	5,662	8,219
A&F transfer	127/243	.703	4,953	7,133
smolagents	131/243	.684	15,181	16,878

5.4 TOOLSANDBOX TRANSFER

The original frozen ToolSandbox matrix is unfavorable to compact routing (Table 7). The basic compact router loses 14.81 success points to Full. Relative to the basic compact router, adding the state graph does not improve success and adds 52.70% end-to-end tokens. On state-dependency trials, it succeeds on 0/39, compared with 15/39 for Full and 27/39 for smolagents. Smolagents obtains 131/243 successes overall but uses 128.74% more end-to-end tokens than Full; its 25 JSON parsing errors remain failures.

The transferred schedule scores 127/243, improving by 29 successes over the always-compact state graph and remaining nine below Full. The paired difference from Full is -3.70 points, CI $[-12.35, +4.94]$, so non-inferiority is not established. Agent-model tokens fall by 6.50%; end-to-end savings are 3.33% with CI $[-7.14, 13.03]$%, which includes zero. A&F obtains 16/39 state-dependency successes and 26/48 multi-user-turn successes, against the state graph's 0/39 and 18/48 and Full's 15/39 and 25/48. These overlapping memberships are descriptive diagnostics, not additional independent tests. Four infrastructure errors remain counted as failures. Transfer thus supports partial recovery in point estimate, not confirmed success retention or an end-to-end efficiency gain.

6 DISCUSSION AND LIMITATIONS

Identifying the scheduling contribution. The main result concerns the complete scheduling-and-routing system. Isolating event placement requires random, periodic, confidence-based, and wider-frontier controls with a shared frozen router and prior. Schedules should be calibrated on development data and compared by realized schema and total-model tokens: equal schema counts need not imply equal costs.

Reliability and transfer. The five-point aggregate margin does not establish equivalence or task-level safety. Scenario overlap, prior dependence, and inconclusive transfer results call for independent on-policy evaluation with and without the prior; optimistic-history replay cannot replace it.

Deployment costs and pilot scope. Model-token reductions do not measure latency, dollar, or accelerator savings. Router computation, demonstration-generation cost, and deployment-specific cache reuse require separate measurement. The larger-catalog pilot uses four development examples, a bounded-broad variant, and changed serving settings; its approximately 81% token savings accompany mixed success retention (Appendix G). Repeats do not create independent tasks, distractors returning errors do not model every irrelevant tool, and deterministic exposure does not imply deterministic model execution.

7 CONCLUSION

A&F alternates broad schema context at observable checkpoints with capped state-conditioned frontiers. Its predeclared BFCL condition comparison meets a five-point aggregate success margin with 15.85% fewer model tokens; a later same-family analysis gives a similar operating point. The combined system thus merits further evaluation with isolated anchor-placement controls and independent scenarios.

REPRODUCIBILITY STATEMENT

The experimental protocol records frozen method and model identities, prompts, proposal caps, next-tool-prior checksums, evaluator settings, expected examples, and statistical seeds. Section 4 and Appendices B–E describe selection, inference, and artifact boundaries; Appendix G separates the later pilot's serving configuration. Full executable reproduction additionally requires the pinned runtime, models, and benchmark data under their upstream licenses; manuscript sources alone are not an experimental reproduction package.

ETHICS STATEMENT

Schema exposure is a context-management decision, not an authorization mechanism. Hiding a tool definition should not substitute for execution permissions, argument validation, or human oversight. Our aggregate success tolerance does not establish safety for individual tasks; deployment in consequential settings requires task-specific reliability and cost assessment. The released material should respect the licenses and privacy requirements of the underlying models and benchmarks.

AI USE STATEMENT

In this work, generative AI tools were used to refine hypotheses and experimental methodology, implement methods and diagnostic scripts, reformat data, check statistical summaries, interpret results, identify related literature, and organize, edit, and format the manuscript. AI-assistant suggestions were not treated as benchmark labels or measured outcomes: reported observations come from benchmark execution records. Generated code was checked through tests and execution. The authors are responsible for the final text, numerical claims, and artifacts.

REFERENCES

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216v3*, 2024. URL https://arxiv.org/abs/2402.03216v3.

Jia-Kai Dong, I-Wei Huang, Chun-Tin Wu, and Yi-tien Tsai. ETOM: A five-level benchmark for evaluating tool orchestration within the MCP ecosystem. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 1453–1488, 2026. doi: 10.18653/v1/2026.findings-eacl.75. URL https://aclanthology.org/2026.findings-eacl.75/.

Wei Fang and James R. Glass. Beyond single-shot: Multi-step tool retrieval via query planning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42119–42144, 2026. doi: 10.18653/v1/2026.findings-acl.2090. URL https://aclanthology.org/2026.findings-acl.2090/.

Uria Franko. Dynamic system instructions and tool exposure for efficient agentic LLMs. *arXiv preprint arXiv:2602.17046*, 2025. URL https://arxiv.org/abs/2602.17046.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11143–11156, 2024. doi: 10.18653/v1/2024.findings-acl.664. URL https://aclanthology.org/2024.findings-acl.664/.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. ToolkenGPT: Augmenting frozen language models with massive tools via tool embeddings. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1988. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8fd1a81c882cd45f64958da6284f4a3f-Abstract-Conference.html.

Xinyi Hong, Pinjun Dong, Xinyang Yu, and Binyan Jiang. Tools are not islands: Set-level tool retrieval for LLM agents via query-conditioned hyperedge prediction. *arXiv preprint arXiv:2607.25718*, 2026. URL https://arxiv.org/abs/2607.25718.

Sijia Li, Yuchen Huang, Zifan Liu, Zijian Li, Jingjing Fu, Lei Song, Jiang Bian, Jun Zhang, and Rui Wang. Experience-evolving multi-turn tool-use agent with hybrid episodic-procedural memory. *arXiv preprint arXiv:2512.07287v3*, 2026. URL https://arxiv.org/abs/2512.07287v3.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Elias Lumer, Anmol Gulati, Vamse Kumar Subbiah, Pradeep Honaganahalli Basavaraju, and James A. Burke. MemTool: Optimizing short-term memory management for dynamic tool calling in LLM agent multi-turn conversations. *arXiv preprint arXiv:2507.21428*, 2025. URL https://arxiv.org/abs/2507.21428.

Bhrij Patel, Davide Belli, Amir Jalalirad, Maximilian Arnold, Aleksandr Ermolov, and Bence Major. Dynamic tool dependency retrieval for lightweight function calling. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33649–33672, 2026. doi: 10.18653/v1/2026.findings-acl.1680. URL https://aclanthology.org/2026.findings-acl.1680/.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. *arXiv preprint arXiv:2305.15334*, 2023. URL https://arxiv.org/abs/2305.15334.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.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 *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.16789.

Vyzantinos Repantis, Ameya Gawde, Harshvardhan Singh, and Joey Blackwell II. How many tools should an LLM agent see? A chance-corrected answer. *arXiv preprint arXiv:2605.24660*, 2026. URL https://arxiv.org/abs/2605.24660.

Furkan Sakizli. Tool-schema compression enables agentic RAG under constrained context budgets. *arXiv preprint arXiv:2605.26165*, 2026. URL https://arxiv.org/abs/2605.26165.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Xin Xie, Dongyun Xue, Wuguannan Yao, Mingxiao Feng, Wengang Zhou, Xiang Qi, Houqiang Li, and Peng Zhang. SGA-MCTS: Decoupling planning from execution via training-free atomic experience retrieval. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 1176–1193, 2026. doi: 10.18653/v1/2026.findings-acl.60. URL https://aclanthology.org/2026.findings-acl.60/.

Hanwen Xu, Xuyao Huang, Yuzhe Liu, and Zhijie Deng. TPS-Bench: Evaluating AI agents' tool planning & scheduling abilities in compounding tasks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34949–34961, 2026. doi: 10.18653/v1/2026.acl-long.1614. URL https://aclanthology.org/2026.acl-long.1614/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

A Controlled synthetic mechanism test

The frozen test comprises 3,000 examples from seeds 142–144 over 10,000 tools, including 2,250 tasks with two to four steps. The structured profile router was frozen before generation. Step decomposition is oracle-provided and execution is deterministic. Table 8 therefore measures routing and state carry under the generator, not autonomous multi-step agent competence. The stateful structured router solves every multi-step task (Wilson 95% interval [.9983,1.0000]); the stateless version deteriorates after the first step.

Table 8: Frozen 10,000-tool mechanism test. Success entries are means over three seeds. Tokens are four-step schema-prompt input tokens, not end-to-end model-token totals.

Method	1 step	2 steps	3 steps	4 steps	Tokens
Structured, stateful	1.000	1.000	1.000	1.000	1,290
Structured, stateless	1.000	.025	.004	.003	1,290
Reranker, stateful	.345	.128	.055	.013	1,290
Schema RAG, stateful	.105	.009	.003	.000	18,982

B Development selection and width sensitivities

The three state-ledger modes were jointly declared on 96 primary examples; Full scores 54/96. The fixed success-first selection rule chose the minimal ledger (Table 9). All modes are retained, including the alternatives that did not win selection.

Table 9: Declared A&F development selection on 96 primary examples. The success difference is relative to Full and expressed in percentage points.

State-ledger mode	Success	Difference	Token saving
Minimal (selected)	53/96	-1.04	14.35%
None	52/96	-2.08	14.96%
Verbose	51/96	-3.13	9.55%

Table 10 preserves the retrospective compact-width comparison. The $K = 12$ run took place after the 27B condition outcomes were opened. Its favorable point estimate is a sensitivity result, not a replacement confirmation or an independently selected optimum.

Table 10: 27B condition-set width sensitivity. Full and $K = 16$ reuse the frozen confirmation. Token counts are means per task.

Exposure policy	Success	Mean tokens	Saving	Evidence role
Full schema	258/394	88,679	—	Frozen baseline
A&F, $K = 16$	259/394	74,622	15.85%	Predeclared
A&F, $K = 12$	259/394	68,767	22.45%	Retrospective

C ADDITIONAL BFCL CONTROLS AND AN ILLUSTRATIVE TRAJECTORY

At $K = 8$, TF–IDF, BGE-M3, and the basic compact router score 36/394, 26/394, and 40/394, respectively. These are schema-count sensitivities rather than cap-matched controls for $K = 16$. Smolagents receives the complete schema set through its standard prompt and is an external full-catalog agent.

The primary Full score of 259/394 (Table 5) and the A&F condition score of 259/394 (Table 3) come from different splits. They must not be subtracted or interpreted as a task-level recovery comparison. Table 11 instead gives one explicitly opened-development example to illustrate the schedule. Its three policies use the official executor and Qwen3.5-27B. It is neither independent confirmation nor a budget-matched anchor experiment.

Table 11: Opened-development example `multi_turn_base_27`. Tokens are per-trajectory model tokens. The compact and A&F rows have different exposure budgets.

Exposure policy	Decision allocation	Success	Tokens
Full schema	12 broad	1	69,361
Always compact	12 compact	0	40,905
A&F	3 user + 1 error + 8 compact	1	60,141

D LEARNED SEQUENCE AND TRANSITION ANALYSES

D.1 FROZEN NEXT-TOOL POLICIES

The 216 examples in shards 7–15 exclude all development-distillation shards. Full, the untrained compact state graph, a 12-example policy, and a development-distilled policy score 135, 86, 83, and 108/216. Distillation improves success over the untrained controller by 10.19 points [3.24,17.13], but remains 12.50 points below Full $[-19.91, -5.56]$. Both policies are reported, including the unsuccessful short policy. These policy-confirmation results are distinct from the later 394-example condition-holdout comparison of the complete A&F system.

D.2 PRIOR-FREE ROUTING REPLAY

The CPU replay supplies reference calls as successful observable history to both versions of the exact frozen router. Recall is gold-tool inclusion at compact decisions. Complete-example coverage requires every evaluated compact decision to include its gold tool. Neither metric is on-policy task success (Table 12).

Table 12: Post-hoc exact-router replay. "Complete" reports covered/evaluated examples with and without the prior. The scenario-disjoint replay uses primary shards 11–15.

Population	Decisions	Prior recall	No-prior recall	Complete (prior / no prior)
Condition holdout	795	96.23%	85.53%	293/317 vs. 221/317
Scenario-disjoint	276	97.10%	82.61%	86/94 vs. 56/94

The candidate width is 16 for both replay variants. Prior-only rescues outnumber prior-free-only rescues 89 to 4 on the condition set and 42 to 2 on the scenario-disjoint set. Cluster-bootstrap intervals for the prior-free-minus-prior complete-coverage difference are $[-29.25, -16.04]$ and $[-46.81, -17.02]$ percentage points. One condition example contains a compact reference call to a tool absent from the officially evolving catalog at that turn; the row is retained and excluded from both recall denominators. Because reference history is optimistic and fixed, these differences cannot determine what a prior-free executing agent would actually do.

D.3 SAME-BUDGET TYPED TRANSITIONS

On opened primary shards 11–15, typed $K = 12$, untyped $K = 12$, A&F $K = 16$, and Full score 80, 76, 77, and 81/120. Typed minus untyped is $+3.33$ points [0,$+7.50$]. The retrospective complete-condition comparison reverses the point estimate: typed scores 256/394 versus 259/394 for untyped and uses 3.26% more tokens [1.35,5.25]%. This mixed development result is not part of the main claim.

E STATISTICAL AND REPRODUCIBILITY DETAILS

BFCL comparisons cluster by numeric scenario identity and use 20,000 percentile-bootstrap replicates. The 27B confirmation uses seed 20260801; the 122B exhaustive analysis uses seed 20260802. The five-point success margin and minimum ten-percent token reduction precede their relevant outcomes. ToolSandbox resamples 81 scenarios while retaining trial seeds 42, 43, and 44. Bootstrap intervals are reported at 95%. Token-saving intervals recompute Equation 3 within each resampled paired set rather than averaging per-task percentage savings.

Configuration and evidence snapshots. The frozen snapshot contains configurations, scripts, reports, and tables, and records model revisions, source hashes, method artifacts, expected identities, executor settings, and token components. Its aggregation protocol rejects missing or duplicate result rows and checks component totals. The main end-to-end results use that evidence; the later catalog pilot and prior-free CPU replay are reported separately, not pooled with confirmation.

Experimental artifact boundary. The paper-writing snapshot excludes runtime source, raw benchmark traces, model weights, datasets, caches, and cluster logs. Full reproduction requires these components under their upstream licenses. The exact cross-encoder checkpoint, fusion/backfill procedure, dependency extraction, prior features, and adapter event handling must come from the pinned runtime; Algorithm 1 specifies only the exposure policy. Manuscript sources are not an executable reproduction package.

Experiment-name correspondence. "Basic compact router" and "Compact router + state graph" correspond to the archived labels "Basic ExplicitLM" and "ExplicitLM + state graph," respectively. These descriptive renamings change neither methods nor outcomes and do not assert reproduction of an external architecture. TF–IDF, BGE-M3, Full, and smolagents labels are unchanged.

Controlled comparisons. Main Full/A&F pairs share the model, decoding budget, state ledger, and executor. Priors and budgets are not interchangeable across the primary controls, condition controller, and ToolSandbox parent router. A schedule-by-prior on-policy experiment, budget-matched anchor placement, and cross-family confirmation remain untested; neither reference-history replay nor cross-split comparisons supply them.

F SECONDARY DIAGNOSTIC BOUNDARIES

Two attempted extensions did not yield confirmatory evidence. The auxiliary 15-example Qwen3.5-122B calibration (the `tau3` sticky-cycle setting) scores 10/15 for A&F and 9/15 for Full, using 12.56% fewer agent tokens. It misses the predeclared 80% overall and domain gates, leaving all 263 frozen tasks sealed. On 54 high-identifiability StableToolBench diagnostic examples, a calibrated sequence gate scores .630 [95% bootstrap CI .500,.759], compared with .407 for available-order planning and .444 for a dependency heuristic. Those diagnostics omit the complete official virtual-execution and evaluator stack and are not official task-success measurements.

G EXPLORATORY REPEATED CATALOG PILOT

Purpose and status. Separate from confirmation, this pilot evaluates four development examples, one per multi-turn category, selected by fixed hash ordering in shards 0–10 before inspecting pilot outcomes. Two catalogs, three policies, two model sizes, and two greedy repeats yield 96 trajectories over only four distinct tasks; no significance or non-inferiority claim is made.

Catalogs and policies. Native catalogs retain official schemas. The 300-tool condition interleaves them with seeded distractors from other benchmark classes. Native names, including later-announced tools, are reserved against collisions; later tools become available only upon official observable announcement. Generation and scoring both return uniform unknown-function errors for distractor calls without changing state. This differs from irrelevant tools that execute successfully.

Full exposes all available schemas. Static-16 freezes the initial 16-tool frontier and appends subsequently announced tools, so its width can exceed 16. Native A&F restores the native catalog at anchors. At 300 tools, its broad pool is capped at 64 tools and 16,384 schema tokens, with the frozen 16-tool router between anchors: this is a bounded-broad variant, not full-catalog restoration.

Execution and accounting. Both Qwen3.5-9B and Qwen3.5-27B use pinned BF16 checkpoints, vLLM 0.19.1, greedy decoding, and a 131,072-token limit. Each policy cell runs four concurrent trajectories; prefix caching resets between cells and policy order reverses in repeat 2. Seeds 42 and 43 identify repeats, not independent randomized samples. The official BFCL executor and scorer determine success. Accepted responses are retained before metering assertions; token IDs and usage are checked against the engine and harness. Context overflows count as failures with prior token charges retained. An evaluator argument-binding error was fixed by CPU rescoring existing Full outputs, without regeneration or duplicate charges. These changed serving settings are not pooled with confirmation.

Table 13: Completed exploratory pilot. Success pools two repeats of four examples. Savings use aggregate model-token totals relative to Full; Static-16 and A&F have different realized budgets.

Model	Catalog	Full	Static-16	A&F	A&F saving
9B	Native	2/8	0/8	2/8	0.53%
9B	300	2/8	0/8	3/8	81.85%
27B	Native	6/8	0/8	5/8	11.63%
27B	300	5/8	1/8	4/8	81.29%

Table 14: Every pilot cell by repeat. Token totals include every accepted model request across both repeats, including failed trajectories. Repeats reuse the same four examples.

Model	Catalog	Policy	Repeat 1	Repeat 2	Total tokens
9B	Native	Full	1/4	1/4	731,066
9B	Native	Static-16	0/4	0/4	584,949
9B	Native	A&F	1/4	1/4	727,189
9B	300	Full	1/4	1/4	5,501,158
9B	300	Static-16	0/4	0/4	317,174
9B	300	A&F	2/4	1/4	998,308
27B	Native	Full	3/4	3/4	806,845
27B	Native	Static-16	0/4	0/4	416,183
27B	Native	A&F	2/4	3/4	713,018
27B	300	Full	2/4	3/4	5,085,105
27B	300	Static-16	1/4	0/4	299,668
27B	300	A&F	2/4	2/4	951,360

Interpretation. At 300 tools, token savings exceed 81% at both model sizes but success retention is mixed: the 9B gain in repeat 1 does not recur in repeat 2, and pooled 27B success is one below Full. A&F uses about 3.2 times Static-16's tokens, so recovery over that policy cannot isolate anchor placement. Independent scenarios and budget-comparable schedules remain necessary.

Execution variability. A serving diagnostic compares isolated and concurrently submitted requests at one active sequence. After 200 isolated references per model, token IDs and finish reasons match in 6/8 initial paired states at 9B and 7/8 at 27B. Both tests stop at the first failing group, missing the planned 200/200 criterion. This is not task-success evidence and does not support bitwise reproducibility.