NEURAL INFORMATION PROCESSING SYSTEMS

TIER: Trajectory-Invariant Execution Rewards for Multi-Step Tool Composition

Anay Kulkarni[*1], **ChiaEn Lu**[*1], Dheeraj Mekala[1], Jayanth Srinivasa[2], Gaowen Liu[2], Jingbo Shang[1]

[1]UC San Diego [2]Cisco Research [*]Equal contribution

NeurIPS 2026 • github.com/anaykulkarni/TIER

Many tasks need several tool calls chained together

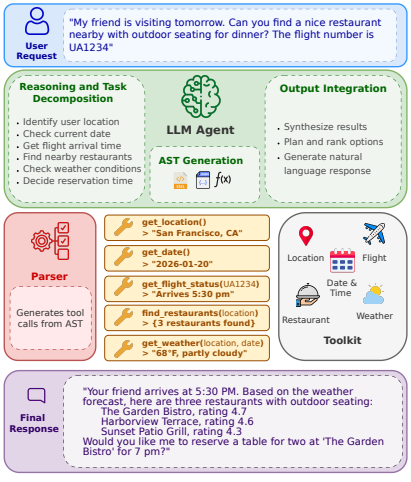
Multi-step tool composition

“What is the weather at the destination of my next flight?”

1. Look up the flight
2. Read off its destination
3. Only then, query the weather

The minimum number of tool calls a task needs is its **depth**; a depth-*d* task is a ***d*-step** task.

Each call depends on an earlier call’s output, so one early mistake breaks the rest of the chain.



NEURAL INFORMATION PROCESSING SYSTEMS

RL with a simple outcome reward collapses with depth

Qwen3-8B trained with a format + final-answer reward; DepthBench accuracy (%)

Steps	0	1	2	3	4	5	6
Simple outcome reward	98.24	**93.12**	**7.05**	**4.67**	**0.00**	**0.00**	**0.00**

> The model learns single calls, but accuracy falls from **93%** at 1 step to **7%** at 2 steps and **0%** from 4 steps onward.

The bottleneck is the **reward**. One right/wrong signal at the end gives no credit to the intermediate choices a long chain depends on: which tool, which arguments, which order.

NEURAL INFORMATION PROCESSING SYSTEMS

Existing reward designs: two failure modes

What a reward for tool composition needs

- **Outcome rewards** score only the final answer. Every failure gets the same score, so there is no signal about *which* step went wrong.
- **Trajectory-supervised rewards** (e.g., ToolRL) score similarity to one annotated reference path. They penalize equally valid paths, need annotation, and must be re-annotated when tools change.

	Path invariance	Graded feedback	No reference path
Outcome reward	✓	✗	✓
Trajectory-supervised	✗	✓	✗
TIER (ours)	✓	✓	✓

Path invariance: every sequence that passes all checks gets the maximum reward, whatever order or equivalent tools it uses. **Graded feedback**: failed sequences are scored by which checks they still pass.

TIER: rewards from schemas and execution

Reward *what the sequence accomplishes*, not *which path it follows*

$$R_{\text{total}} = R_{\text{format}} + R_{\text{parse}} + R_{\text{exec}} + R_{\text{answer}}$$

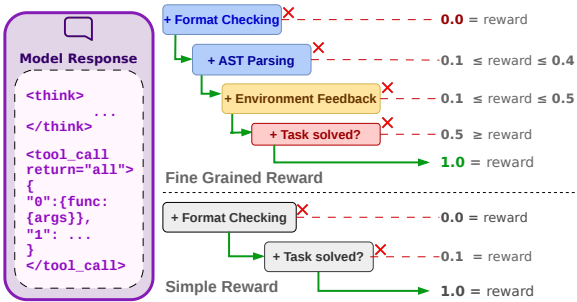
scaled to $[0, 1]$, computed once per generated tool-call sequence

Format	$\{0, 1\}$	The tool calls, written as an AST, are well formed. If not, every other term is 0.
Schema	$[0, 3]$	All tool names are valid; each parameter or type mismatch against the schema costs 0.25.
Execution	$\{0, 1\}$	Every call executes. A failure at step k breaks the calls that depend on it.
Answer	$\{0, 5\}$	The final answer is correct. Weighted highest so the model cannot stop at valid-looking calls.

No term uses a reference trajectory: the verifier reads the tool schemas and runs the calls.

Graded and path-invariant at the same time

Fine-grained TIER reward vs. a simple outcome reward



- **Path invariance.** Each term is scored once over the whole sequence, so the maximum reward does not depend on length or sub-goal order.
- **Graded feedback.** A failed sequence is scored by the checks it still passes.
- **Tools can change.** The verifier reads the schemas, so new or changed tools need no re-annotation.

DepthBench

A depth-stratified benchmark for multi-step tool composition

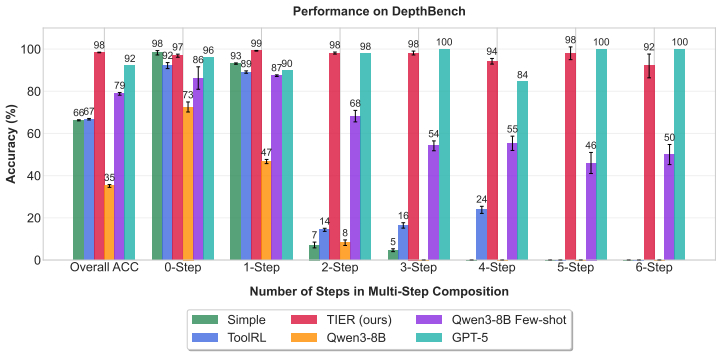
- **163 executable tools** with deterministic backends, modeled on common APIs.
- **1,710 hand-crafted queries** from depth 0 (no tool applies) to depth 6.
- Chains $A \to B \to C$, funnels $(A, B) \to C$, trees $A \to (B, C)$, and order-agnostic structures.
- **944 train / 766 validation** with disjoint target tools and distractors. Distractors share schemas with the targets.

Depth	0	1	2	3	4	5	6
Validation queries	51	480	95	75	45	10	10

Training: GRPO-style policy gradient on Qwen3-8B, 8 rollouts per prompt. Baselines train the same model on the same data with a simple outcome reward or ToolRL's reward.

TIER stays above 90% at every depth

DepthBench validation accuracy, mean of 10 evaluation runs

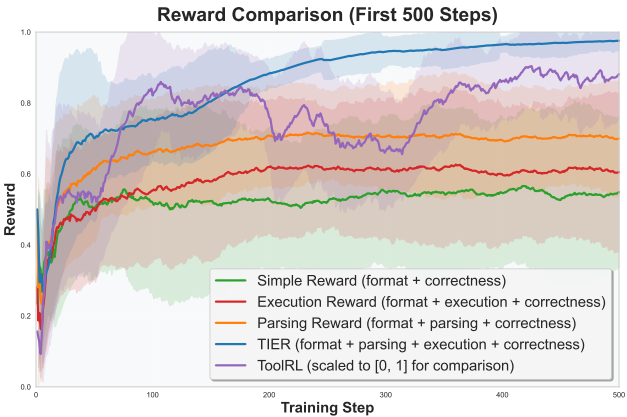


- **TIER**: **98.39%** overall, at least 92% at every depth.
- **ToolRL**: below 25% from 2 steps onward, 0% at 5–6 steps.
- **Untrained, 3 examples**: 46–68% on 2- to 6-step tasks.

GPT-5 (3-shot) is a frontier reference only: a proprietary model, not comparable in scale to the 8B models we train.

Only the full reward trains stably

Mean training reward under each reward design

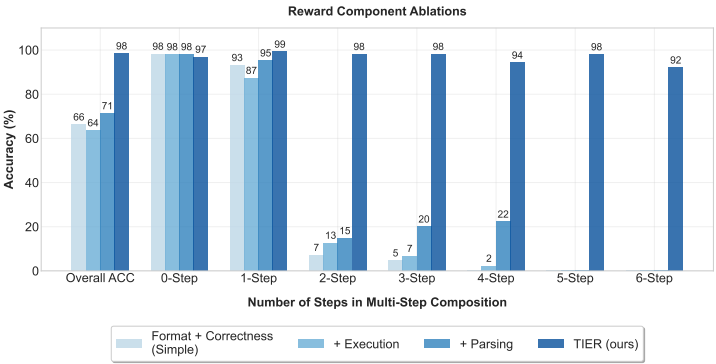


- **TIER** converges steadily to high reward.
- **Partial rewards** have wide variance bands and plateau early at a lower value.
- **ToolRL** oscillates without settling: its supervision is inconsistent across alternative valid paths.

ToolRL's reward is rescaled from $[-3, 4]$ to $[0, 1]$.

Every reward level is needed

Removing the schema or execution term



- Without the schema or execution term, accuracy stays at or below **22%** from 2 steps onward.
- Only the full reward learns deep compositions.

NEURAL INFORMATION PROCESSING SYSTEMS

Gains transfer to other benchmarks

Same Qwen3-8B models, single runs

BFCL v3 (no demonstrations)

	Overall	Multiturn
TIER	**68.92**	**39.12**
Simple reward	66.30	34.00
SFT	61.47	19.25
ToolRL	37.27	0.38
Qwen3-8B, our prompt	63.20	28.25
Qwen3-8B, default prompt	64.31	33.38

Prompting alone stays near 64. Under the same prompt, TIER lifts Qwen3-8B from 63.20 to 68.92, most on Multiturn.

NestFUL (exact match)

No demonstrations	
TIER	**0.684**
ToolRL	0.476
3-shot setup of Basu et al.	
TIER (8B)	**0.75**
DeepSeek-V3 (685B)	0.60
GPT-4o	0.60

Takeaways

- Reward design is the bottleneck for multi-step tool composition.
- TIER rewards what a sequence *achieves*, checked by schemas and execution: every valid path gets full credit, and failures still get graded feedback.
- No reference trajectories, so tools can change without re-annotation.
- TIER stays above 90% at every depth on DepthBench, and the gains carry over to BFCL v3 and NestFUL.

Next: multi-turn RL, other verifiable domains (code, NL-to-SQL), and a depth curriculum.

Code, data, and paper
github.com/anaykulkarni/TIER