Agentic Context Engineering (ACE): Self-Improving LLMs via Evolving Contexts, Not Fine-Tuning

October 10, 2025

TL;DR: A team of researchers from Stanford University, SambaNova Systems and UC Berkeley introduce ACE framework that improves LLM performance by **editing and growing the input context** instead of updating model weights. Context is treated as a living "playbook" maintained by three roles—**Generator, Reflector, Curator**—with small **delta items** merged incrementally to avoid brevity bias and context collapse. Reported gains: **+10.6%** on AppWorld agent tasks, **+8.6%** on finance reasoning, and **~86.9% average latency reduction** vs strong context-adaptation baselines. On the AppWorld leaderboard snapshot (Sept 20, 2025), **ReAct+ACE (59.4%) ≈ IBM CUGA (60.3%, GPT-4.1)** while using **DeepSeek-V3.1**.

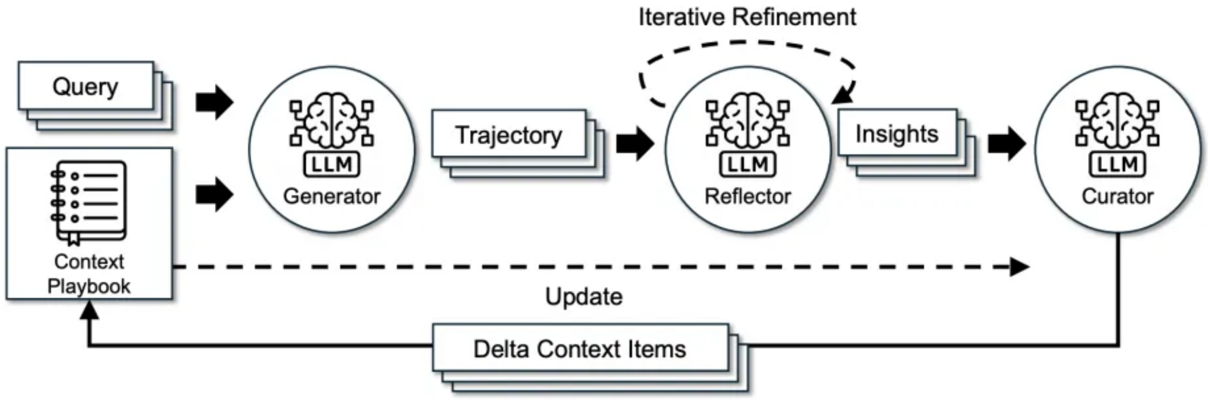

Figure 4: **The ACE Framework.** Inspired by Dynamic Cheatsheet, ACE adopts an agentic architecture with three specialized components: a Generator, a Reflector, and a Curator.

https://arxiv.org/pdf/2510.04618

What ACE changes?

ACE positions “context engineering” as a first-class alternative to parameter updates. Instead of compressing instructions into short prompts, ACE **accumulates and organizes domain-specific tactics** over time, arguing that higher **context density** improves agentic tasks where tools, multi-turn state, and failure modes matter.

Method: Generator → Reflector → Curator

- **Generator** executes tasks and produces trajectories (reasoning/tool calls), exposing helpful vs harmful moves.
- **Reflector** distills concrete lessons from those traces.

- **Curator** converts lessons into typed **delta items** (with helpful/harmful counters) and merges them deterministically, with de-duplication and pruning to keep the playbook targeted.

Two design choices—**incremental delta updates** and **grow-and-refine**—preserve useful history and prevent "context collapse" from monolithic rewrites. To isolate context effects, the research team fixes **the same base LLM (non-thinking DeepSeek-V3.1)** across all three roles.

Benchmarks

AppWorld (agents): Built on the official ReAct baseline, **ReAct+ACE** outperforms strong baselines (ICL, GEPA, Dynamic Cheatsheet), with **+10.6% average** over selected baselines and **~+7.6%** over Dynamic Cheatsheet in online adaptation. On the **Sept 20, 2025 leaderboard**, **ReAct+ACE 59.4% vs IBM CUGA 60.3% (GPT-4.1)**; ACE **surpasses CUGA** on the harder **test-challenge** split, while using a smaller open-source base model.

Finance (XBRL): On **FiNER** token tagging and **XBRL Formula** numerical reasoning, ACE reports **+8.6% average** over baselines with ground-truth labels for offline adaptation; it also works with execution-only feedback, though quality of signals matters.

Method	GT Labels	Test-Normal		Test-Challenge		Average
		TGC↑	SGC↑	TGC↑	SGC↑	
DeepSeek-V3.1 as Base LLM						
ReAct		63.7	42.9	41.5	21.6	42.4
Offline Adaptation						
ReAct + ICL	✓	$64.3_{+0.6}$	$46.4_{+3.5}$	$46.0_{+4.5}$	$27.3_{+5.7}$	$46.0_{+3.6}$
ReAct + GEPA	✓	$64.9_{+1.2}$	$44.6_{+1.7}$	$46.0_{+4.5}$	$30.2_{+8.6}$	$46.4_{+4.0}$
ReAct + ACE	✓	$\mathbf{76.2}_{+12.5}$	$\mathbf{64.3}_{+21.4}$	$\mathbf{57.3}_{+15.8}$	$\mathbf{39.6}_{+18.0}$	$\mathbf{59.4}_{+17.0}$
ReAct + ACE	✗	$75.0_{+11.3}$	$\mathbf{64.3}_{+21.4}$	$54.4_{+12.9}$	$35.2_{+13.6}$	$57.2_{+14.8}$
Online Adaptation						
ReAct + DC (CU)	✗	$65.5_{+1.8}$	$\mathbf{58.9}_{+16.0}$	$52.3_{+10.8}$	$30.8_{+9.2}$	$51.9_{+9.5}$
ReAct + ACE	✗	$\mathbf{69.6}_{+5.9}$	$53.6_{+10.7}$	$\mathbf{66.0}_{+24.5}$	$\mathbf{48.9}_{+27.3}$	$\mathbf{59.5}_{+17.1}$

Table 1: **Results on the AppWorld Agent Benchmark.** "GT labels" indicates whether ground-truth labels are available to the Reflector during adaptation. We evaluate the ACE framework against multiple baselines on top of the official ReAct implementation, both for offline and online context adaptation. ReAct + ACE outperforms selected baselines by an average of 10.6%, and could achieve good performance even without access to GT labels.

https://arxiv.org/pdf/2510.04618

Method	GT Labels	FINER (Acc↑)	Formula (Acc↑)	Average
DeepSeek-V3.1 as Base LLM				
Base LLM		70.7	67.5	69.1
Offline Adaptation				
ICL	✓	$72.3_{+1.6}$	$67.0_{-0.5}$	$69.6_{+0.5}$
MIPROv2	✓	$72.4_{+1.7}$	$69.5_{+2.0}$	$70.9_{+1.8}$
GEPA	✓	$73.5_{+2.8}$	$71.5_{+4.0}$	$72.5_{+3.4}$
ACE	✓	$\mathbf{78.3}_{+7.6}$	$\mathbf{85.5}_{+18.0}$	$\mathbf{81.9}_{+12.8}$
ACE	✗	$71.1_{+0.4}$	$83.0_{+15.5}$	$77.1_{+8.0}$
Online Adaptation				
DC (CU)	✓	$74.2_{+3.5}$	$69.5_{+2.0}$	$71.8_{+2.7}$
DC (CU)	✗	$68.3_{-2.4}$	$62.5_{-5.0}$	$65.4_{-3.7}$
ACE	✓	$\mathbf{76.7}_{+6.0}$	$76.5_{+9.0}$	$\mathbf{76.6}_{+7.5}$
ACE	✗	$67.3_{-3.4}$	$\mathbf{78.5}_{+11.0}$	$72.9_{+3.8}$

Table 2: **Results on Financial Analysis Benchmark.** "GT labels" indicates whether ground-truth labels are available to the Reflector during adaptation. With GT labels, ACE outperforms selected baselines by an average of 8.6%, highlighting the advantage of structured and evolving contexts for domain-specific reasoning. However, we also observe that in the absence of reliable feedback signals (*e.g.*, ground-truth labels or execution outcomes), both ACE and other adaptive methods such as Dynamic Cheatsheet may degrade, suggesting that context adaptation depends critically on feedback quality.

https://arxiv.org/pdf/2510.04618

Cost and latency

ACE's **non-LLM merges** plus localized updates reduce adaptation overhead substantially:

- **Offline (AppWorld): −82.3% latency** and **−75.1% rollouts** vs **GEPA**.
- **Online (FiNER): −91.5% latency** and **−83.6% token cost** vs **Dynamic Cheatsheet**.

Method	GT Labels	Test-Normal		Test-Challenge		Average
		TGC↑	SGC↑	TGC↑	SGC↑	
DeepSeek-V3.1 as Base LLM						
ReAct		63.7	42.9	41.5	21.6	42.4
Offline Adaptation						
ReAct + ACE w/o Reflector or multi-epoch	✓	70.8 $_{+7.1}$	55.4 $_{+12.5}$	55.9 $_{+14.4}$	38.1 $_{+17.5}$	55.1 $_{+12.7}$
ReAct + ACE w/o multi-epoch	✓	72.0 $_{+8.3}$	60.7 $_{+17.8}$	54.9 $_{+13.4}$	39.6 $_{+18.0}$	56.8 $_{+14.4}$
ReAct + ACE	✓	76.2 $_{+12.5}$	64.3 $_{+21.4}$	57.3 $_{+15.8}$	39.6 $_{+18.0}$	59.4 $_{+17.0}$
Online Adaptation						
ReAct + ACE	✗	67.9 $_{+4.2}$	51.8 $_{+8.9}$	61.4 $_{+19.9}$	43.2 $_{+21.6}$	56.1 $_{+13.7}$
ReAct + ACE + offline warmup	✗	69.6 $_{+5.9}$	53.6 $_{+10.7}$	66.0 $_{+24.5}$	48.9 $_{+27.3}$	59.5 $_{+17.1}$

Table 3: **Ablation Studies on AppWorld.** We study how particular design choices of ACE (iterative refinement, multi-epoch adaptation, and offline warmup) could help high-quality context adaptation.

Method	Latency (s)↓	# Rollouts↓
ReAct + GEPA	53898	1434
ReAct + ACE	9517 $_{(-82.3\%)}$	357 $_{(-75.1\%)}$

(a) **Offline** (AppWorld).

Method	Latency (s)↓	Token Cost ($)↓
DC (CU)	65104	17.7
ACE	5503 $_{(-91.5\%)}$	2.9 $_{(-83.6\%)}$

(b) **Online** (FiNER).

Table 4: **Cost and Speed Analysis.** We measure the context adaptation latency, number of rollouts, and dollar costs of ACE against GEPA (offline) and DC (online).

https://arxiv.org/pdf/2510.04618

Key Takeaways

- **ACE = context-first adaptation:** Improves LLMs by incrementally editing an evolving "playbook" (delta items) curated by Generator→Reflector→Curator, using the *same* base LLM (non-thinking DeepSeek-V3.1) to isolate context effects and avoid collapse from monolithic rewrites.
- **Measured gains:** ReAct+ACE reports **+10.6%** over strong baselines on AppWorld and achieves **59.4%** vs **IBM CUGA 60.3% (GPT-4.1)** on the Sept 20, 2025 leaderboard snapshot; finance benchmarks (FiNER + XBRL Formula) show **+8.6%** average over baselines.
- **Lower overhead than reflective-rewrite baselines:** ACE reduces adaptation latency by **~82–92%** and rollouts/token cost by **~75–84%**, contrasting with Dynamic Cheatsheet's persistent memory and GEPA's Pareto prompt evolution approaches.

Conclusion

ACE positions context engineering as a first-class alternative to weight updates: maintain a persistent, curated playbook that accumulates task-specific tactics, yielding measurable gains on AppWorld and finance reasoning while cutting adaptation latency and token rollouts versus reflective-rewrite baselines. The approach is practical—deterministic merges, delta items, and long-context–aware serving—and its limits are clear: outcomes track feedback quality and task complexity. If adopted, agent stacks may "self-tune" primarily through evolving context rather than new checkpoints.

Asif Razzaq is the CEO of Marktechpost Media Inc.. As a visionary entrepreneur and engineer, Asif is committed to harnessing the potential of Artificial Intelligence for social good. His most recent endeavor is the launch of an Artificial Intelligence Media Platform, Marktechpost, which stands out for its in-depth coverage of machine learning and deep learning news that is both technically sound and easily understandable by a wide audience. The platform boasts of over 2 million monthly views, illustrating its popularity among audiences.