AUTOMATION ANYWHERE

Technical White Paper

Building Better Agents: τ-bench #1 Performance, GBA-Bench Evaluation, and PRE + Context Intelligence

Matt Wright, Henry Lee, Sandra Wang, Emily Gal

This paper presents two complementary bodies of research. The first is external validation: Automation Anywhere's goal-based agents achieved the highest scores overall in our τ-bench evaluation—the industry-standard agent benchmark from Princeton and Sierra—at every pass level, achieving 74.5% at pass^1 with a consistent ~4-point lead over all leaderboard competitors. These results were achieved by aa-agent-v1 with lightweight context scaffolding (embedding-based memory retrieval and mid-conversation course correction) and reflect the strength of our underlying evaluation framework and agent architecture. The second body of research asks: can we do better? Using our proprietary Process Reasoning Engine (PRE) and two novel memory architectures, we demonstrate that adding context intelligence to enterprise agents produces goal completion lifts of up to 32% and trajectory accuracy improvements of 20–47% on GBA-Bench, our in-house enterprise evaluation suite. Together, these results describe a complete methodology: rigorous evaluation, benchmark-validated performance, and a research roadmap for continuous improvement.

Key Contributions:

- #1 on τ-bench overall at every pass level (pass^1–pass^4), 375 tasks, 4 domains — using aa-agent-v1 with lightweight context scaffolding.
- Up to 3.2× faster than leaderboard competitors across domains.
- GBA-Bench: proprietary enterprise evaluation suite covering 7 domains, 30+ frontier models benchmarked.
- PRE + Context Intelligence: up to 32% goal completion uplift and 20–47% trajectory accuracy gain on goal based agents.
- Practical guidance on memory architecture, quality filtering, and deployment conditions.

These results were submitted to the τ-bench public leaderboard in May 2026 (PR #303, pending merge at time of publication). All performance comparisons reference published leaderboard scores at the time of submission.

1. Introduction

1.1 The Evaluation Framework: A Quick Recap

In our previous white paper, A Framework for Evaluating Goal-Based AI Agents, we established that a single success metric is not sufficient for enterprise automation. Two agents can produce the same final outcome via very different execution paths—one following an optimal, auditable workflow, another guessing its way through redundant tool calls and fragile reasoning. We called this the 'Scrappy Win,' and showed it appeared in over 40% of runs on complex agent types.

Our response was a dual-metric evaluation framework: Task Success (did the agent complete the business objective?) and Trajectory Accuracy (did it do so through the correct reasoning path?). We use an LLM-as-judge evaluator calibrated against τ-bench human labels with 0.99 agreement. This infrastructure is the foundation for everything that follows.

1.2 Two Stories, One Paper

This paper tells two sequential stories. The first is benchmark performance: aa-agent-v1, with lightweight context scaffolding, ranks #1 on τ-bench overall at every pass level. This validates the quality of our underlying agent architecture and evaluation methodology against the best publicly available standard.

The second story is improvement research: we then asked whether agents augmented with structured operational memory—the Process Reasoning Engine (PRE) with Context Intelligence—perform meaningfully better on our enterprise-specific benchmarks. The answer, tested on GBA-Bench across four enterprise domains, is a clear yes: 20–47% trajectory accuracy gains, up to 32% goal completion uplift, and a 5× improvement on our most challenging workflow.

These are distinct research tracks. τ-bench measures aa-agent-v1. PRE + Context Intelligence is an additional layer that we are actively developing and that will ship as part of our Context Intelligence product. We present both because together they describe a complete picture: where we stand today, and where we are going.

> **Important Distinction**
>
> The τ-bench #1 ranking was achieved by aa-agent-v1 — a lightweight context scaffolding system with embedding-based memory retrieval and mid-conversation course correction. PRE + Context Intelligence is separate research, tested on GBA-Bench evaluation suite. We present both to show the full methodology: strong foundations, then a principled path to making agents even better.

2. τ-bench: #1 at Every Pass Level

2.1 What τ-bench Measures

τ-bench is the industry-standard agent evaluation benchmark developed by Princeton and Sierra. It evaluates agents on multi-turn tasks requiring accurate tool selection, correct parameter construction, and coherent state management across 375 tasks spanning four real-world service domains: Airline, Retail, Telecom, and Banking.

τ-bench uses pass^k scoring: to receive credit at pass level k, the agent must complete the task correctly on k independent consecutive runs. Pass^1 measures raw accuracy. Pass^4 measures reliability—the agent must be correct four times in a row, with no opportunity to get lucky. This structure directly tests the property that matters most for enterprise deployment: consistent, repeatable correctness under varied inputs.

2.2 Results: Consistent Lead Across All Pass Levels

aa-agent-v1 ranks #1 overall at every pass level, pooled across all 375 tasks and four domains. Table 2.1 shows the full pass-level breakdown against the next-best system.

Pass Level	aa-agent-v1	Leaderboard #1	Delta
pass^1	74.5%	70.2%	**+4.3 pts**
pass^2	67.9%	63.1%	**+4.8 pts**
pass^3	63.6%	59.3%	**+4.3 pts**
pass^4	60.3%	56.2%	**+4.1 pts**

Table 2.1: τ-bench pass-level results — aa-agent-v1 vs. Top published leaderboard score at time of submission (May 2026), pooled across 375 tasks and 4 domains.

The ~4-point lead reproduces at pass^2, pass^3, and pass^4. This is not an accuracy artifact that decays under repeated evaluation — it is a structural advantage that holds up as the bar for reliability rises. Pass^4 in particular is a demanding standard: an agent must execute the same task type correctly on four independent runs, each from a fresh state. A consistent ~4-point lead through pass^4 confirms that our agents are fundamentally more reliable, not just marginally more accurate on a given run.

> **Why Pass^k Matters for Enterprise Deployment**
>
> A production agent executes the same workflow type hundreds of times per day across different inputs. Pass^1 tells you what happens once. Pass^4 tells you what you can count on every time. Our consistent ~4-point lead through pass^4 means our agents are not occasionally correct — they are reliably correct.

2.3 Domain Breakdown: Accuracy and Speed

The overall ranking is composed of strong individual domain results. Speed is as important as accuracy for production deployment: an agent that takes 12 minutes per task cannot run at enterprise scale. Table 2.2 shows both dimensions.

Domain	AA Score	vs. Leader	Execution Speed	Rank
Airline	84.5%	**+0.5 pts**	1.6× slower	#1
Retail	82.9%	−1.5 pts	3.2× faster	#2
Telecom	98.2%	**+0.4 pts**	2.6× faster	#1
Banking	31.7%	**+0.5 pts**	2.7× faster	#1

Table 2.2: τ-bench results by domain — accuracy and execution speed vs. leaderboard leader. (May 2026)

Across all four domains, aa-agent-v1 shows either accuracy or speed leadership, with Telecom and Banking showing both. The Airline domain is the speed exception: our submission runs slower than the leaderboard, a trade-off attributable to memory retrieval and mid-conversation course correction steps in aa-agent-v1. Retail trails the accuracy leader (Qwen3.5-397B-A17B at 84.4%) by 1.5 points but leads on speed by 3.2×. Both Retail accuracy and Airline execution time are priority areas for the next evaluation cycle.

Banking deserves specific attention. The absolute scores are low across all competitors (our 31.7% is still #1) because retrieval latency is the binding constraint in that domain — not model reasoning quality. The agent must retrieve policy and account information in real time, and that retrieval is slow. This is precisely the problem that Context Intelligence is designed to address: by pre-fetching and structuring relevant operational context before the agent starts executing, we reduce dependence on real-time retrieval mid-task. Banking is therefore the τ-bench domain where we expect the greatest future improvement as Context Intelligence matures.

3. GBA-Bench: Evaluation Built for Enterprise

3.1 Why Public Benchmarks Alone Are Insufficient

τ-bench is a rigorous and respected public standard, and performing well on it matters. But public benchmarks have structural limitations for enterprise AI deployment: they test general-purpose agent tasks rather than enterprise-specific workflows; they do not include domain policy or compliance validation; and their task sets cannot be extended to cover the schemas, tools, and business rules of a specific organization.

An agent that scores well on τ-bench is a strong foundation. An agent additionally validated against real enterprise workflows—with proprietary test coverage—is a production-ready system. We built the ladder.

3.2 GBA-Bench: What It Is and How It Works

GBA-Bench is our proprietary evaluation suite for goal-based agents running backend enterprise workflows. It covers seven domains: Banking, Insurance, Healthcare, Supply Chain, Sales, Finance, and Vendor Onboarding.

Test cases are generated from real source documents: SOPs, support tickets, and workflow definitions. A four-stage pipeline converts those documents into structured agent definitions, scenario-milestone pairs, and executable Python test classes. New models can be evaluated against the full suite in hours rather than days, which means when a customer asks about a newly released model, we already have the numbers.

To date, we have formally evaluated 30+ frontier models spanning every major model family: Anthropic, OpenAI, Google, Meta, Qwen, DeepSeek, Mistral, and Zhipu/GLM. Table 3.1 shows a representative leaderboard snapshot, reported separately across Task Success and Trajectory Accuracy to preserve the distinction between outcome and reasoning quality.

Rank	Model	Task Success	Trajectory Accuracy
01	claude-opus-4.5	0.912	0.810
02	claude-opus-4.6	0.910	0.757
03	claude-haiku-4.5	0.903	0.764
04	claude-sonnet-4.5	0.914	0.740
05	claude-sonnet-4	0.812	0.713
06	qwen3-235	0.832	0.673
07	gpt-4o / o4-mini / claude-4.1	0.69–0.75	0.59–0.68
30+	gemini · mistral · glm-4.5…	0.14–0.80	0.08–0.59

Table 3.1: GBA-Bench leaderboard snapshot May 2026 — avg. Task Success + Trajectory Accuracy across enterprise scenarios. 30+ frontier models evaluated.

GBA-Bench evaluation uses the same dual-metric framework described in our prior work: Task Success (did the agent complete the business objective?) and Trajectory Accuracy (did it follow the correct reasoning path?). The LLM-as-judge evaluator achieves 0.99 trajectory agreement against τ-bench human labels. A win on one metric does not constitute a passing result—both scores are required.

> **The Proprietary Evaluation Moat**
>
> Public benchmarks are table stakes—every serious AI team runs them. GBA-Bench is proprietary IP: real enterprise scenarios built from actual SOPs and workflows, domain policy validation, and a pipeline that generates new test cases from any customer's own documentation. This evaluation infrastructure compounds over time as we add domains, models, and agent types.

4. PRE + Context Intelligence: Making Agents That Learn

4.1 The Problem: Stateless Agents Repeat Mistakes

Even well-performing base agents have a structural limitation: they start every task run with no memory of previous executions. The same tool parameter errors recur. The same incorrect sequences produce the same Scrappy Wins. The same valid-but-inefficient paths waste API calls and latency. This statelessness is not a model quality problem—it is an architectural one, and it is fixable.

The cost is most visible on complex, multi-step workflows. On our Customer Churn Prevention agent, baseline trajectory accuracy without memory is 0.12—only 12% of runs follow a correct reasoning path. The agent frequently reaches a plausible-looking outcome through an incorrect sequence of tool calls: a Scrappy Win at scale. This is the gap Context Intelligence closes.

4.2 The PRE Memory Architecture

Our Process Reasoning Engine (PRE) provides the infrastructure for structured operational memory. We investigated two classes of memory architecture, both compared against a no-memory baseline.

The first approach uses a single consolidated memory store. After each agent run completes, a compressed summary is generated and stored alongside its Trajectory Accuracy and Goal Completion scores. On subsequent runs, semantic search retrieves the most relevant past experiences and injects them into the system prompt before execution begins. A quality-based merging step ensures the memory store improves over time: perfect executions (Trajectory = 1.0 AND Goal = 1.0) are preserved separately and intact, while imperfect runs are distilled into 3–5 high-impact lessons rather than stored verbatim.

The second approach separates memory into two distinct tiers: strategy-level and procedural-level. Strategy memory captures high-level Thought → Tool → Observation → Conclusion workflow patterns, retrieved at the start of each run to set execution context. Procedural memory captures granular state-transition records—what tools were just called, the current agent state, the reasoning, and the outcome—retrieved mid-task after each tool execution to inform the agent's next decision.

The key innovation in the dual-tier approach is how procedural queries are generated. Rather than re-using the original user prompt as the search query—which becomes stale once the agent is mid-task—queries are constructed from the tools just executed: 'Observation: Retrieved data from

{tools}. State: Information from {tools} available.' This grounds retrieval in the agent's current operational state rather than its starting point. Only perfect executions are retrieved as procedural examples, preventing the agent from learning incorrect sequences from imperfect runs.

4.3 Experimental Setup

Both memory approaches were evaluated against a no-memory baseline across four enterprise agent types from GBA-Bench. All three configurations used identical underlying models—frontier language models for agent execution and context summarization, and a text embedding model for semantic memory retrieval—to ensure comparability across conditions.

Agent Type	Business Role
Claim Details	Insurance claims: retrieve and verify claim, policy, and police report details
Customer Churn Prevention	SaaS retention: detect at-risk accounts and recommend targeted interventions
Finance Credit Hold	Credit hold resolution: investigate root cause, validate docs, orchestrate approvals
Sales Deal Acceleration	Q-end deal closer: identify blockers, coordinate approvals, send stakeholder alerts

Table 4.1: GBA-Bench Agent Types Used in Context Intelligence Experiments

4.4 Results: Trajectory Accuracy

Trajectory accuracy improves across every agent type when context learning is enabled. Table 4.2 shows baseline versus PRE + Context Intelligence results.

Agent Type	Baseline (no memory)	With PRE+CI	Improvement (pp)
Claim Details	0.70	0.90	**+0.20**
Customer Churn Prevention	0.12	0.59	**+0.47**
Finance Credit Hold	0.35	0.55	**+0.20**
Sales Deal Acceleration	0.33	0.66	**+0.33**

Table 4.2: Trajectory Accuracy May 2026 — Baseline vs. PRE + Context Intelligence (GBA-Bench)

The trajectory accuracy improvement ranges from 20 to 47 percentage points across domains, with Customer Churn Prevention showing a ~4.9× lift (0.12 → 0.59). These are the same agent types our prior research identified as having the highest Scrappy Win rates. Context learning directly targets the root cause: agents that previously reached correct outcomes through incorrect reasoning paths now follow validated, purposeful execution sequences.

4.5 Results: Goal Completion and Tool Efficiency

Goal completion improves by up to 32 percentage points with PRE + Context Intelligence enabled, most prominently on Customer Churn Prevention and Sales Deal Acceleration. The Claim Details agent shows a ceiling effect—its baseline goal completion is already high, and context learning adds limited marginal value in that regime.

A secondary benefit is tool call efficiency. On Sales Deal Acceleration, context-enabled agents reduce average tool calls per run by roughly 20%, reflecting correct first-attempt behavior rather than error-and-retry cycles. In production this translates directly to lower API cost, reduced latency, and more predictable execution time.

> **The Core Story**
> Agents with PRE and Context Intelligence aren't just more successful—they're more trustworthy. Task success tells you the agent got the answer. Trajectory accuracy tells you it got there the right way. Memory improves both, but it's the trajectory gain that converts a Scrappy Agent into a Safe Agent that a business can rely on.

4.6 Qualitative Example: Eliminating a Recurring Failure

On the Sales Deal Acceleration agent, the baseline runs consistently showed the same failure: the agent called send_deal_alert with alert_type: 'contract_acceleration'—a value not in the tool's accepted enum. It received an error, then retried with the correct value 'escalation'. Task succeeded; trajectory degraded. A textbook Scrappy Win.

With context learning enabled, the agent retrieved a procedural memory that distilled this pattern into the explicit lesson: 'Always verify and use valid alert types when sending escalation or meeting notifications.' On the next run, it invoked send_deal_alert correctly on the first attempt. Trajectory score: 1.0. No retry. The agent did not need to encounter the failure again—it consulted the lesson from a prior run's distilled experience and acted on it immediately.

5. The Connection: Why the Same Methodology Drives Both Results

The τ-bench leadership and the PRE + Context Intelligence gains are distinct experiments, but they share the same underlying logic. τ-bench primarily rewards agents that achieve the correct final outcome, which depends on effective tool use, minimal redundancy, and maintaining coherent state across turns. These outcomes are strongly influenced by the agent's underlying decision-making process—precisely the behaviors that context learning helps improve.

aa-agent-v1, which scores #1 overall on τ-bench, is strong because our evaluation framework is calibrated to detect and penalize the failure modes τ-bench also penalizes—Scrappy Wins, invalid tool calls, circular reasoning. The PRE + Context Intelligence layer takes those same failure modes and eliminates them through accumulated operational memory, producing further gains on GBA-Bench enterprise tasks.

Banking on τ-bench is the clearest forward-looking connection. Our Banking domain score (31.7%) is constrained by retrieval latency. Context Intelligence is specifically designed to reduce real-time retrieval dependence by pre-structuring relevant operational experience before the agent begins executing. As PRE + context learning is applied to τ-bench configurations in Phase 2, Banking is the domain where we expect the most visible improvement.

6. Deployment Guidance: Getting the Most from Context Intelligence

6.1 Evaluation Is a Prerequisite, Not an Add-On

Context learning is only as reliable as the evaluation signals it depends on. Every memory is stored with its Trajectory Accuracy and Goal Completion scores. These scores determine whether a memory is preserved as a replicable success pattern or distilled as a cautionary lesson. Without accurate evaluation, the memory store accumulates noise and agents learn from—and repeat—their own failures.

This means GBA-Bench evaluation and PRE + Context Intelligence are designed as an integrated system. Do not deploy context learning without a validated evaluation pipeline running on every agent run. The evaluation infrastructure is not overhead—it is the signal that makes memory learning safe.

6.2 Memory Volume and the Bootstrapping Phase

When fewer than approximately 10 memories are stored, retrieved context is more likely to be weakly relevant or misleading than genuinely helpful. The dual-tier architecture gates procedural memory retrieval behind a 10-memory threshold precisely to avoid premature activation. Context learning should be understood as a compound benefit: the first 50–100 agent runs are a bootstrapping phase. Significant, reliable gains become observable once a sufficient corpus of quality-filtered memories has accumulated.

6.3 Where Context Intelligence Helps Most

Based on Phase 1 results, the strongest gains occur for agents with these characteristics:

- Baseline Trajectory Accuracy below 0.70 — substantial room for behavioral correction.
- Tool parameter variability — valid parameter values are non-obvious and discovered through failure.
- Repeated task structures — the same workflow type executed with different inputs, enabling genuine pattern accumulation.
- Multi-step decision chains — early incorrect choices cascade into degraded downstream trajectories.

Context learning provides minimal marginal benefit for near-ceiling tasks (Claim Details baseline trajectory: 0.70, goal completion already high) or for single-tool tasks with no structured decision pattern to learn. Teams should prioritize deployment on agents with high observed Scrappy Win rates or low trajectory accuracy.

7. Conclusion

aa-agent-v1 ranks #1 on τ-bench at every pass level, with a consistent ~4-point lead over all leaderboard competitors across pass^1 through pass^4. In three of four domains we also lead on speed—2.5× to 3.2× faster (Airline is the speed exception at 1.6× slower). These results reflect the strength of our underlying agent architecture and evaluation framework, enhanced by lightweight context scaffolding.

When we add PRE + Context Intelligence—structured, quality-filtered memory of past agent executions—performance improves further. Goal completion lifts up to 32%, trajectory accuracy improves 20–47% across GBA-Bench enterprise domains, and our most challenging workflow (Customer Churn Prevention) shows a ~4.4× trajectory gain. These are not the same experiment as τ-bench; they are the next step beyond it.

Three conclusions:

- **Strong foundations matter: We evaluated our agents against τ-bench, achieving the highest scores across all pass levels compared to all results published at the time of our submission (May 2026)**. The evaluation framework that produced those agents—dual-metric, trajectory-aware, calibrated against human labels—is the foundation for everything that follows.
- **Evaluation and improvement are inseparable:** Context learning is not separate from evaluation—it depends on it. Every memory requires an accurate quality signal to be useful rather than harmful. Real-time evaluation and PRE + Context Intelligence are an integrated system, not independent features.
- **The methodology compounds:** The combination of public benchmark leadership (τ-bench), proprietary enterprise evaluation (GBA-Bench), and a principled memory-learning layer (PRE + CI) constitutes a methodology that compounds over time. Each new model evaluation, each new domain, each additional memory cycle makes the system stronger.

The agents that will define enterprise automation are not simply the most accurate on a given benchmark—they are the ones that get the right answer reliably, efficiently, and in a way that improves with every run. That is what this methodology is built to produce.