Agent Performance Analysis Report

What we're delivering: A structured analysis of your two primary agents (**Outreach Agent** and **Onboarding Agent**) grounded in real execution data, including full agent prompts, tool inventories, scenario breakdowns, findings with trace-level evidence, and recommendations targeting specific prompt sections.

What we need from you: Feedback on accuracy, context about your architecture and iteration process, and alignment on priorities so we can move from analysis to validated improvements.

How We Analyzed

Your team manages prompts via Langfuse's UI (enabling non-technical team members to iterate on prompts directly), uses the Vercel AI SDK as the agent framework, and operates an operator/executor pattern with custom judges evaluating agent sessions on a 0–5 scale. Each of your ~40–50 event types has its own trigger prompt that instructs the agent with a specific mission while the system prompt stays constant.

Data Window & Sources

Our analysis covers a **24-hour production window** totaling **254,265 records** and **7.4 GB** exported from Langfuse.

Cost figures and percentages in this report come from three source datasets at different scopes:

Dataset	Scope	Traces	Used For
Full 24h export	All agents, all versions	7,866 total	System overview, daily/monthly cost projections, system-wide QA stats
Agent A v1 discovery profile	Agent A short prompt only	4,197	Tool usage rates, path frequencies, gate percentages, agent-specific QA rate
Agent B discovery profile	Agent B v1+v2 analyzed subset	635	Per-trigger tool usage, QA rejection rates per scenario, agent-specific metrics

Where findings-level analysis (deep trace inspection, savings estimates) is cited, the sample is smaller — noted per finding. All percentages state which dataset they derive from.

Sampling Strategy

Analysis Type	Sample Size	Agent	Method
Scenario classification	300 executions	Agent A (v1 + v2/v3)	Stratified sample across event types
QA rejection taxonomy	200 executions (55 rejections)	Agent A (primarily v1)	Random sample with rejection enrichment
Deep root-cause analysis	17 Agent A + 8 Agent B executions	Both agents	Selected for diversity: send/skip, cheap/expensive, clean/rejected

Agent B QA rejection analysis	635 executions (349 rejections)	Agent B	Complete sample of analysis window
Tool performance profiling	500 executions	Agent A (v1 + v2)	Random sample for statistical stability

---

System Overview

Metric	Value
Estimated monthly AI spend (all agents)	**~$12,500**
Agent A's share	**91.6%** (~$11,500/month)
Daily Agent A executions	6,312
Model	Claude Haiku 4.5 (99.7% of calls)
Cache hit rate	68.2%
Response latency (p90)	11.8 seconds

Agent A: Outreach Agent

What it does: When new job matches are found for existing contacts, the agent decides whether to notify them and crafts the outbound message. Event-driven, one-shot automation with no human interaction. Every action passes through a QA gate (a separate LLM call) before execution.

Operational Profile

Stat	v1 (Short Prompt)	v2/v3 (Long Prompt)
Daily executions	4,197 (66.5%)	2,100 (33.5%)
Avg cost per execution	$0.064	Higher (32 tools, 7x larger prompt)
Send rate	43.3%	73.1%
QA rejection rate	**22.3%**	Lower (different scenario mix)

Decision Graph: Expected vs Observed

We mapped the expected agent behavior (from the system prompt) against what actually happens in production traces. Key divergence: the prompt prescribes a 2-gate skip path, but we observed a 3-gate pattern where most skips happen *after* expensive data fetching — not before.

The 15 most common execution paths cover 48.5% of all traces:

Rank	Traces	Outcome	Path (abbreviated)	Cost
1	391 (9.3%)	SKIP	Ctx → List → Details+ → Review → Memory	$0.065

2	195 (4.6%)	**SEND**	Ctx → List → Details+ → Review(✓) → Draft → Send → Memory	$0.092
3	181 (4.3%)	SKIP	Ctx → List → 1xDetails → Review	$0.033
4	172 (4.1%)	SKIP	Ctx → Review (quick skip)	$0.018
7	135 (3.2%)	**SEND**	Ctx → Details+ → Draft → Review(✕) → Draft → Review(✓) → Send	$0.111

Findings

Finding 1: Formatting Violations Are the #1 QA Rejection Category

QA rejection share: 30.9% of all rejections

The draft generation step doesn't have access to formatting rules, so it regularly produces drafts with em dashes and spaced hyphens. The QA gate rejects them. The agent rewrites. Often multiple times.

Rejection Category	Count	%
Formatting (dashes)	**17**	**30.9%**
Policy violation (hard dates)	10	18.2%
Preference mismatch (geo/salary/visa)	7	12.7%
Fabrication (hallucinated claims)	7	12.7%
Link violation (missing/wrong URLs)	3	5.5%
Forbidden word ("submit", etc.)	3	5.5%
Other	8	14.5%

Cost of rejections: Executions with at least one rejection average **8.28 LLM calls** vs **5.21 without** — a 59% compute increase. Each rejection cycle adds ~20 seconds.

Finding 2: Fabrication Follows a Predictable Pattern

QA rejection share: 12.7% of all rejections

The agent invents facts not returned by any tool — most commonly visa sponsorship availability, salary ranges (misquoted by small amounts), and fabricated URLs.

Evidence from a specific trace: The agent fetched 4 opportunity details — none mentioned visa sponsorship. The agent composed instructions claiming *"opportunities that specifically offer visa sponsorship support."* The QA gate caught it, but the agent rephrased the same fabrication 5 times before removing it. **71% of this execution's tokens were wasted.**

Finding 3: Skip-Path Executions Over-Fetch Data

Scale: The 10 most common execution paths are all skip paths.

More than half of outbound notifications end in a skip. But in many cases, the agent fetches 3–6 detailed records *before* making the skip decision. The signal to skip was available after the first 3 context calls.

Skip Segment	% of v1 Executions	Avg Cost	What's Happening
Cheap skip (<$0.03)	~25%	$0.019	Decided early, minimal waste
Normal skip ($0.03–0.07)	~12%	$0.048	Over-fetched before deciding
Expensive skip (>$0.07)	**~5%**	**$0.096**	Fetched 4–6 details, then skipped

The 4 unnecessary detail calls in expensive skips add ~$0.050 per execution and ~12 seconds of latency — for zero output.

Recommendations

Grouped by implementation type — we separated what can be done in the prompt from what needs engineering.

Prompt Improvements

#	Recommendation	Target Section	Impact
1	Add a no-fabrication pre-draft checklist	`<critical_rules>`	Reduce fabrication rejections (12.7% of QA load)
2	Elevate formatting rules to highest-compliance section	`<critical_rules>`	Reduce formatting rejections (30.9% of QA load)
3	Add deliverability pre-check	`<execution_flow>`	Prevent wasted executions on unreachable contacts
4	Add safe alternative phrasings for banned language	`<voice>`	Reduce forbidden-word rejections
5	Strengthen date-reference rule	`<critical_rules>`	Reduce date-based rejections (18.2% of QA load)

Architectural Improvements

#	Recommendation	Scenarios Affected	Proposed Change
6	Skip-before-fetch gate	Outbound notifications (v1)	Enforce skip decision after context calls, before detail fetching
7	Pre-invocation routing filter	All scenarios	Add string-match filter to catch misrouted events before agent invocation

Code-Level Improvements

#	Recommendation	Proposed Change

8	Retry limit on message delivery	Limit to 1 retry. On second failure: log, update memory, schedule a retry reminder. Stop.
9	Pre-QA formatting regex	Add regex check after draft output: catch dashes, forbidden words. Fix automatically. Zero LLM cost.

Agent B: Onboarding Agent

What it does: Handles all incoming interactions from first contact through onboarding to matchmaking and ongoing support. Responds to 8 different trigger types. Same operator/executor pattern and QA gate as Agent A, but with a single large prompt shared across all scenarios.

Operational Profile

Stat	Value
Share of total spend	**8.0%** ($33.37/day, ~$1,001/month)
Daily executions	1,204
Avg cost per execution	$0.028
QA rejection rate	**66%** first-attempt failure

Key Finding: Sign-Off Bug Drives 58% of Rejections

Three contradicting instructions across three prompt sections cause the QA gate to reject every draft that includes a sign-off name:

Location	What It Says	Effect
`<voice>` line 39	"use the recruiter name from the conversation thread"	Agent uses name from channel context
`<voice>` line 51	"NEVER sign off with 'Leo' or any agent/bot name"	Agent avoids "Leo" but thinks human names are fine
`<critical_rules>`	—	**No sign-off rule exists here** (the highest-compliance section has no guidance)

Combined with the runtime channel context injecting a human name, the agent signs as that name — and the QA gate rejects it every time.

Additional Findings & Recommendations

#	Finding	Impact	Recommendation
1	Sign-off contradiction (3 locations, 3 conflicting rules)	58% of rejections	Consolidate sign-off rules into `<critical_rules>`
2	Unbounded context retrieval (returns ALL messages)	Cost scales linearly with conversation length	Add limit parameter

3	Memory recall skipped in 50% of executions	Agent loses context from previous interactions	Make memory retrieval mandatory in `<execution_flow>`
4	Dash rule buried 30+ lines deep	25% of rejections are formatting	Move to `<critical_rules>`
5	Job search tool underused (6 calls in 100 executions)	Instruction buried 120 lines deep in prompt	Move to earlier, higher-visibility section

---

Analysis performed by the Mutagent founding team. March 2026. This is a sanitized version. Company-specific identifiers have been redacted.