GTM-Bench: Evaluating Agentic AI Systems for Buyer/Seller Coherence in Go-to-Market Workflows

Maksymilian Polaczuk*, Jeremy Baron*, Reuben Horne*
Trevor Cavill, Callum Mudgeway, Chris Herrmann
* equal contribution, corresponding authors
Blackpearl Group Limited
research@blackpearl.com

Abstract

AI agents are increasingly deployed in go-to-market (GTM) workflows, where value depends not on generating more outreach but on selecting buyers that coherently match a seller's offer. Existing agent and CRM benchmarks evaluate tool use, business-system operations, or grounded reporting, but do not isolate the end-to-end GTM problem: infer the offer, infer an actionable ideal customer profile, retrieve accounts or contacts, and audit each returned record as a commercially relevant match. We introduce GTM-Bench, the first benchmark for evidence-grounded buyer/seller coherence in GTM workflows. GTM-Bench contains 72 realistic tasks, designed from a taxonomy of real prospecting queries submitted to Bebop.ai. Agents operate in a controlled data and tool environment and produce three artifacts: an offer summary, an ICP, and a ranked prospect list. We evaluate six frontier generalist agent systems and one purpose-built GTM system. Trace analysis shows that successful systems combine broad retrieval with evidence-based pruning, while failures stem from identity errors, unsupported claims, and poor stopping decisions. We release the task catalog, agent harness, evaluation code, benchmark calculation code, and leaderboard to support progress on commercially useful, auditable GTM agents.

1 Introduction

Benchmarks have shaped progress in agentic systems by turning broad claims about intelligence into measurable tasks. Early public benchmarks emphasized text-only question answering, mathematical reasoning, and language understanding of Large Language Models (LLMs) (Hendrycks et al., 2021; Cobbe et al., 2021; Srivastava et al., 2023). More recent benchmarks measure agents that interact with tools, files, terminals, browsers, and business systems (White et al., 2025; Merrill et al., 2026; Jimenez et al., 2024; Zhou et al., 2023; Mialon et al., 2023). This shift matters because many valuable uses of AI are not single-response tasks: they are long-horizon workflows in which a system must inspect an environment, retrieve evidence, and make decisions under uncertainty.

GTM work is one such setting. Sales and marketing teams increasingly use AI systems alongside traditional data platforms such as Clay, Apollo, and ZoomInfo. A typical GTM workflow requires understanding what a seller offers, understanding the buyer accounts and people most likely to need it: Ideal Customer Profile ("ICP"), then retrieving a set of matching profiles, ready for activation in the corresponding sales and marketing campaign. Success of the GTM workflow is largely based on achieving high *buyer/seller coherence*: the system should match the right offer to the right account, at the right time.

This problem is commercially important and socially relevant. Poor GTM matching wastes sales and marketing budgets, but it also creates noise for buyers through irrelevant emails, calls, and advertisements. As AI lowers the marginal cost of outbound generation, benchmarks must distinguish systems that increase relevance from systems that simply scale low-quality prospecting. A benchmark that rewards volume without evidence would encourage exactly the wrong behavior.

Existing enterprise-agent and CRM benchmarks are valuable but leave this gap open. CRMArena and CRMArena-Pro evaluate agents on realistic CRM tasks such as customer service, information retrieval, and

policy compliance (Huang et al., 2024; Huang et al., 2025). Sales Research Bench evaluates an AI sales-research application on grounded text and chart outputs over sales data (Bhol, 2026). These benchmarks test business-system reasoning, but they do not isolate the GTM lead-generation problem: infer the offer, infer the ICP, retrieve and rank accounts or contacts, and audit whether each returned row is a coherent buyer match.

We introduce GTM-Bench, a benchmark consisting of 72 carefully constructed GTM tasks covering common and high-value GTM workflows, inspired by real prospecting behavior from users on the Bebop.ai platform.

We evaluate six frontier generalist agents, including open-source models and harnesses, and a purpose-built GTM system. We open source all 72 questions, agent/harness runner code, evaluation code, and benchmark calculation code. Our codebase is available at github.com/bpg-bebop/gtm-bench, and our leaderboard is available at gtm-bench.ai.

2 GTM-Bench

GTM-Bench evaluates agents on realistic GTM work. Success requires identifying accounts and contacts that are both commercially relevant and timely for outbound selling, with evidence sufficient to justify each recommendation. This section defines the task formulation, dataset, and scoring protocol.

2.1 Task Formulation

A GTM-bench task consists of an instruction, a database environment, and an operating environment. An instruction is a natural language user prompt which specifies the GTM objective. The database environment contains all of the available datasets and tables. The operating environment contains all system prompts, agent skills, Docker container, filesystem and regular agentic tools such as web search and terminal use.

Formally, a GTM-bench task instance is defined by

$$x = (q, D, E),$$

where q is the natural-language user prompt, D is the data environment, and E is the operating environment.

A GTM system A maps the task instance to inferred GTM structure and ranked buyer records:

$$A(x) = A(q, D, E) \Rightarrow (\hat{o}_A, \hat{c}_A, R_A),$$

where $\hat{o}_A$ is the inferred seller offer and $\hat{c}_A$ is the inferred ICP. The final output is a ranked set of buyer records,

$$R_A = \{r_1, \ldots, r_k\}.$$

Each row r_i represents a candidate prospect. Each task must be completed by retrieving the best possible corresponding set R_A. Producing $\hat{o}_A$ and $\hat{c}_A$ is important because these artifacts ground retrieval for R_A and provide context for downstream agentic work, such as campaign generation, email generation, and marketing collateral.

2.2 Source Corpus

User data collected from Bebop.ai enables our benchmark to be based on actual GTM user behavior from real sellers. We constructed a source corpus using a sample of 59,881 opening queries submitted to Bebop.ai, a self-serve GTM platform that maps seller context to potential buyers, from 2025-02-07 through 2026-05-19. These are natural-language prompts from users trying to find customers, leads, or other GTM targets.

We applied an embedding-based clustering analysis of each task in the source corpus, allowing the authors to identify recurring user behavior and design a canonical taxonomy spanning a wide range of GTM tasks.

The resultant query taxonomy is shown in Table 1.

Task	Percentage	Illustrative Example
Domain based lead search	43.8%	"Find customers for xyz.com"
Generic lead search	17.6%	"Find perfect customers seeking b2b lead generation services"
Persona search	6.8%	"find me 100 vp of engineering, cto, head of talent acquisition, engineering manager, and hr director contacts at us-based companies…"
Offer to ICP discovery	6.2%	"i'm building software for ticket management for a number of christian camps... can you help me find customers who can benefit from ticketing software like this?"
Geographic/local search	5.1%	"show me small businesses in california who can use local seo services."
Firmographic filtering	4.5%	"find companies that are growing, revenue $100m-$1b, employee count of 200+ employees, asset intensive, in the united states…"
Industry category search	4.4%	"show me companies in the renewable energy sector"
Technographic search	4.1%	"find businesses using hubspot and salesforce"
Intent/trigger search	3.6%	"find u.s.-based companies… shown strong intent signals in the last 24 hours related to outsourcing, offshoring, or hiring external staffing vendors…"
Lookalike search	3.5%	"find me companies similar to [REDACTED]"
Market research	1.7%	"research market trends for the sales industry"
Non-GTM / Ambiguous	1.4%	"hello"

Table 1: Taxonomy of real-world GTM queries from Bebop.ai. The resulting taxonomy shows that real GTM search behavior is dominated by domain-based lead discovery with a long-tail of prospecting task types.

Turning our taxonomy into useful guidance for our benchmark required discarding low-information, off-topic, and junk prompts from our taxonomy. Ambiguous prompts such as "find me the perfect customers" provided too little information to be assessed and were discarded. Off-topic prompts such as "hello" were another category that were obviously excluded. The authors acknowledge a great degree of sample bias – due to input format on Bebop.ai – towards "domain based lead search" and "generic lead search", so we decided the benchmark distribution should be much flatter across the taxonomy.

Additionally, to ensure our benchmark has representative coverage of verticals, our analysis revealed a wide range of seller/buyer verticals used throughout the corpus. Verticals spanned highly commercial areas such as digital marketing and insurance, to non-commercial, including religious organizations and government.

The specificity and difficulty of prompts varied greatly: incomplete context, underspecified targets, and evidence gaps were frequently included.

2.3 Question Generation and Selection

We originally generated 140 candidate benchmark questions by sampling from the tagged categories of real GTM queries and using GPT-5.5 to draft representative tasks that preserved observed request patterns while remaining evaluable. The candidates were then manually reviewed and accepted or rejected by the authors based on a set of benchmark inclusion criteria.

Selection favored recurring Bebop usage patterns that naturally required identifying accounts and contacts for seller outreach. Prompts were excluded if they were primarily strategic, research-oriented, non-GTM related, or did not require identifying accounts or contacts.

The GTM-bench task set contains 72 selected tasks, spanning 11 task types, and 15 verticals. Table 2 shows the suite composition.

Table 2: GTM-bench task set.

Suite	Task Type	Count	Share	Question IDs
A	Offer-grounded lead lists	6	8.3%	Q001, Q003, Q005, Q006, Q008, Q009
B	Named-domain offer extraction	8	11.1%	Q012, Q013, Q014, Q017, Q018, Q021, Q022, Q024
C	Offer-to-lead list	9	12.5%	Q025, Q028, Q031, Q034, Q035, Q037, Q038, Q041, Q042
D	Vertical / geo / firmographic search	8	11.1%	Q043, Q044, Q047, Q048, Q049, Q050, Q055, Q056
E	Persona / contact activation	9	12.5%	Q058, Q060, Q061, Q063, Q067, Q069, Q070, Q072, Q073
F	Intent / trigger evidence	7	9.7%	Q074, Q075, Q077, Q078, Q081, Q082, Q083
G	Technographic search	5	6.9%	Q086, Q088, Q090, Q092, Q093
H	Offer-grounded lookalikes	3	4.2%	Q096, Q097, Q098
I	Market-to-lead list	3	4.2%	Q102, Q103, Q105
J	Buyer search with contact details	10	13.9%	Q107, Q108, Q110, Q111, Q113, Q114, Q116, Q117, Q119, Q121
K	Limited-context prompts	4	5.6%	Q124, Q126, Q130, Q131

Table 3: Example benchmark questions.

Question ID	Task Type	Prompt
Q025	Offer-to-lead list	I sell SEO/content marketing services. Briefly define the ICP, then generate ranked lead companies in marketing agencies.
Q124	Limited-context prompts	Find decision makers in Phoenix at companies with 51–200 employees showing intent for cybersecurity or cloud migration in the past 30 days.
Q012	Named-domain offer extraction	Find the best-fit customers for bluevine.com. First infer what the company appears to sell, then generate ranked target-company leads that fit the likely ICP.

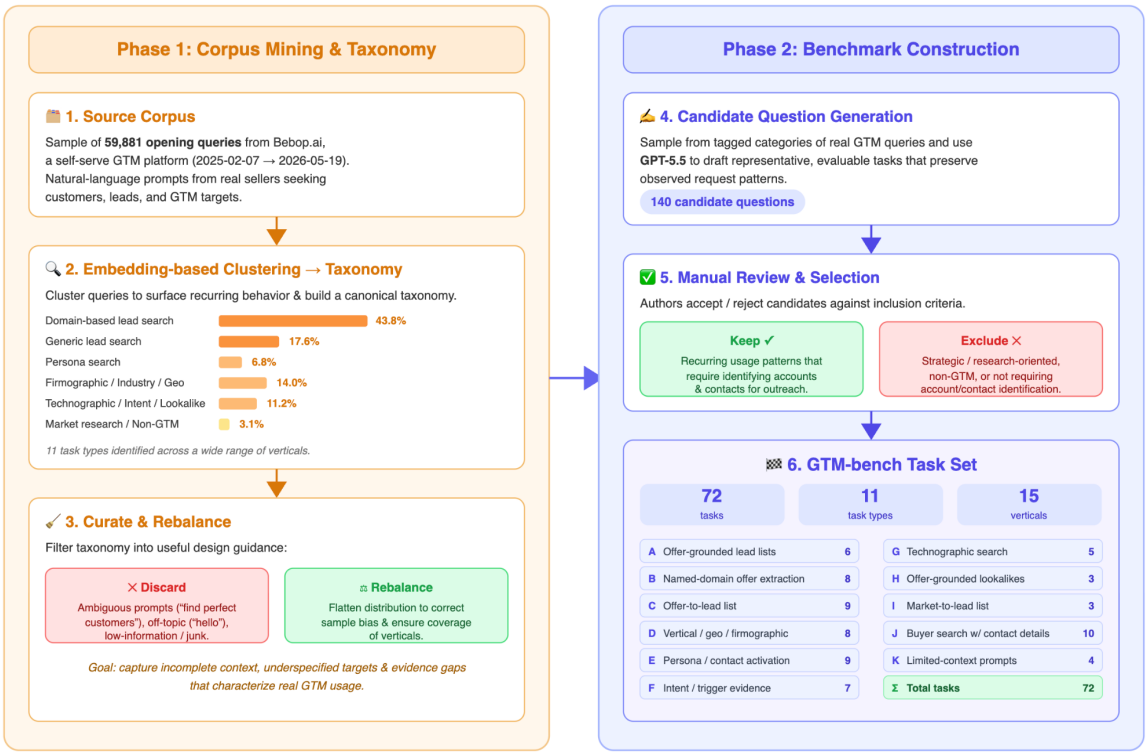

Figure 1: Our dataset construction process requires first sampling from the Bebop.ai queries, generating candidate questions, and reviewing before including in GTM-Bench.

2.4 Environment

Each agent operates in a computing environment that approximates a GTM operator's terminal workspace. The primary structured data interface is a read-only API to Blackpearl's Pearl Engine databases. Agents could access the data environment through a provided detail Skill that specified the approved discovery and retrieval procedures, using the `SKILL.md` format. Additionally, agents were given access to web search, local search, and a sandboxed filesystem for intermediate analysis and outputs.

Excerpt from the provided `SKILL.md` workflow.

```
name: public-data-access
description: Use when a GTM Bench run needs traceable prospecting evidence from
web search, local files, and other explicitly available harness tools.
```

Public Data Access

Use the evidence sources exposed in the current harness to build a defensible lead/prospect artifact. Every final row should be traceable to evidence the evaluator can inspect or reproduce from the run artifacts.

Workflow

...

The data environment was designed to support auditable retrieval without exposing unrestricted backend access. Agents were expected to use approved data sources, cite the evidence used for each returned record, and avoid outputs that depended on inaccessible or unverifiable fields.

The available data covered both company-level and contact-level information, including firmographics, seller-facing business descriptions, contact attributes, quality signals, and observed intent or profile signals. Agents could combine these structured records with indexed website text and open-web evidence when needed.

2.5 Scoring

As per the task definition, in practice agents are instructed to output their results as Markdown and CSV, specifically: $\hat{o}_A$ as `OFFER.md`, $\hat{c}_A$ as `ICP.md`, and R_A as RESULTS.csv. Each artifact is judged independently according to a separate LLM-as-a-judge scoring model, each with their own system prompt, scoring rubric, and input context.

All LLM-as-a-judge models score their respective criteria using a 1-5 scale, where:

- 1 = unusable or wrong
- 2 = weak or materially flawed
- 3 = plausible but generic
- 4 = strong with minor gaps
- 5 = rare near-perfect

Offer Artifact The offer artifact is scored on five dimensions by an LLM judge J_o. J_o assesses `OFFER.md` against the task instruction q.

Criteria	**Reason**
offer-intent fidelity	If the inferred offer drifts from what is actually being sold, every downstream artifact targets the wrong thing.
product specificity	Distinguishing what is being sold from similar or generic offers is necessary to support precise targeting and avoid Buyer/Seller mismatch.
value proposition	Clear and accurate value proposition is necessary to build credible demand and urgency logic into the ICP definition.
concision	Any additional, unnecessary content can only distract from the relevant and useful content.
separation from the ICP	Maintaining a clear distinction between offer and ICP reduces cognitive overhead and avoids Buyer/Seller mismatch

The resulting offer score $Q_o(x)$ is calculated by averaging across each dimension and normalising to [0,1].

$$Q_o(x) = \frac{1}{5|D_o|} \sum_{d \in D_o} J_o(x)_d.$$

Here, D_o is the set of offer criteria and $J_o(x)_d$ is the judged 1–5 score for task x on criterion d.

ICP Artifact $\hat{c}_A$ The ICP artifact is scored on six dimensions by an LLM judge J_c. J_c assesses the `ICP.md` against the offer artifact `OFFER.md` and the task instruction q.

Criteria	Reason
ICP alignment	If the buyer profile doesn't follow from what is sold and why, the resulting lead list targets accounts that have no reason to buy.
Buyer-account specificity	Distinguishing the actual buyer from broad or generic audiences is necessary to support precise account discovery and avoid false positives.
GTM actionability	Without practical filters and exclusions, the ICP cannot actually drive account or contact discovery.
concision	Any additional, unnecessary content can only distract from the relevant and useful content.
separation from the offer	Maintaining a clear distinction between ICP and offer reduces cognitive overhead and avoids Buyer/Seller mismatch.

The ICP score $Q_c(x)$ is derived using the same method as $Q_o(x)$.

$$Q_c(x) = \frac{1}{5|D_c|} \sum_{d \in D_c} J_c(x)_d.$$

Both artifact quality scores, $Q_o(x)$ and $Q_c(x)$, are used as task-level multipliers so that a good row-level output is not rewarded when it is downstream of an incorrect offer or ICP.

Each row from RESULTS.csv, r_i, is evaluated with separate match and audit judge models, J_m and J_a, respectively, and attached context.

Match Judge

The purpose of the match judge J_m is to assess the quality of a candidate output record r_i against the provided task instruction q, associated offer/ICP artifacts (`OFFER.md`, `ICP.md`), and the fully matched database record, website context, and relevant search engine results: achieved with the web search tool. Table 3 outlines the fields that J_m evaluates.

Table 3: Match judge fields.

Criteria	Reason
company fit	If the company itself doesn't match the offer and ICP, the row is a wrong lead regardless of how complete or well-written the rest of it is.
contact fit	Even when company fit is good, a contact without the right role or buying responsibility means outreach lands on someone who can't act on the offer.
offer fit	A concrete reason to need the offer is what separates a real lead from a company that merely fits the vertical on paper.
evidence sufficiency	Without enough credible evidence, the other fit scores are guesses, and a plausible-sounding but unverifiable row could pass as a real lead.
contactability	Missing or unusable details mean a well-matched lead still can't actually be reached.

Audit Judge

LLM's are known to hallucinate. To avoid rewarding a seemingly good record that is fictional or has been embellished by a given agent, we introduce a judge and scoring system to audit for factual accuracy.

The audit judge J_a verifies the factual accuracy of claims and alleged data belonging to the candidate output record r_i, grounded against the fully matched database record, website context, and relevant search engine results.

Table 4 outlines the fields that J_a evaluates.

Table 4: Audit judge fields.

Criteria	Reason
identity resolution	If the record is not linked to a specific entity, a large share of its claims and its relevance as a lead can't be validated.
claim support	Unsupported or embellished claims destroy outreach confidence and are unlikely to refer to real high quality leads.
database consistency	If the output contradicts the authoritative matched record, the agent has altered or invented real data and the record can no longer be trusted.

We opt for a thresholding scoring method instead of continuous probability weighting to reflect the binary nature of GTM work and buyer decisions. In this nature, we introduce a good/neutral/bad scoring system via A & B grade records, and sub-B grade. We consider a record "A Grade", if each applicable field from the match and audit judges J_m and J_a are at least 4: "strong with minor gaps". A record is B-grade if every applicable field is at least 3 but at least one field falls below 4. Records below B-grade receive negative utility.

For a task x, let

$$u(r) = \begin{cases} 1, & r \text{ is A-grade,} \\ 0, & r \text{ is B-grade,} \\ -1, & r \text{ is below B-grade.} \end{cases}$$

The volume-weighted task score is

$$S_A(x) = Q_o(x)Q_c(x) \sum_{r \in R_A(x)} u(r).$$

The benchmark score is the sum over all tasks:

$$S = \sum_x S_A(x).$$

This design deliberately treats low-quality records as harmful. A GTM system cannot score well by returning the full database because every poor fit, unsupported claim, or unresolved identity can reduce total utility. This mirrors a real GTM environment, where there are real costs for not achieving buyer/seller coherence: e.g. wasted sales and marketing spend for the seller, and spam to the 'prospect'.

3 Experimental Setup

3.1 Systems

We evaluated six "generalist" model/harness combinations and one "purpose-built" GTM system on GTM-bench. We chose to evaluate the current frontier model from each major lab: OpenAI, Anthropic and Google, alongside the current frontier open source models: DeepSeek V4 Pro, Kimi K2.6. Each model was evaluated through the agent harness in which it is expected to perform best or most naturally. Codex by OpenAI and Claude Code by Anthropic are highly performant and widely used with their respective models, and Hermes by Nous Research was selected for the remainder, due to its high popularity on OpenRouter at the time of writing. Blackpearl RTSA is a proprietary purpose-built GTM system that combines various LLMs, harnesses, datasets and workflows, which has been adapted for the exact output task contract format. RTSA was configured to record token usage, but unlike the generalist agents, did not expose a trace for qualitative analysis.

Model	Harness and settings
blackpearl-rtsa	Blackpearl RTSA, a purpose-built GTM AI system for offer, ICP, retrieval, and activation.
openai_gpt-5.5	codex-cli + openai:gpt-5.5; medium reasoning.
anthropic_claude-opus-4.7	claude-code-cli + claude-opus-4-7; extra-high effort.
anthropic_claude-sonnet-4.6	claude-code-cli + claude-sonnet-4-6; high effort.
google_gemini-3.5-flash	hermes-agent-cli + gemini-3.5-flash; Hermes default medium reasoning, tool-use enforcement for Gemini/Gemma names.
openrouter_deepseek-v4-pro	hermes-agent-cli + deepseek/deepseek-v4-pro; Hermes default medium reasoning.
openrouter_kimi-k2.6	hermes-agent-cli + moonshotai/kimi-k2.6; Hermes default medium reasoning.

Table 5: Evaluated systems and default run settings.
GPT, Claude, and Gemini models were accessed through their official APIs (platform.openai.com, console.anthropic.ai, aistudio.google.com); Open Source models DeepSeek and Kimi were accessed through OpenRouter.

3.2 Run Protocol

Each agent performed the 72 production tasks with the same expected output contract: `OFFER.md`, `ICP.md`, and RESULTS.csv, representing $\hat{o}_A$, $\hat{c}_A$, and R_A, respectively. For every result record, agents were instructed to preserve canonical database identifiers such as `UP_ID` when available. These identifiers let the evaluator retrieve the corresponding full database record, as described in the match and auditor judge scoring section.

Initial pilot runs that were only given the task instruction q and available tool descriptions produced short, incomplete runs and often fewer than 10 records. To give each agent a fair opportunity, we prepared detailed instructions through a CLAUDE.md instruction file for Claude Code, and `AGENTS.md` for all other agents. `AGENTS.md` explains the scoring incentives and data environment clearly, but is intentionally not prescriptive about any problem solving approaches. The instruction emphasized that every high-quality match is rewarded, every poor match is penalized, data will be checked for accuracy, and complete contact records are more valuable than partial records when contact output is requested.

Excerpt from `AGENTS.md`.

You are GTM Bench, a prospecting agent.

Your objective is to take a specific GTM-related query and fulfill it as effectively as possible, using the tools available in the current harness.

When finding prospects, identify as many correct fits as possible. Each correct fit increases the score; each miss or incorrect prospect decreases the score.

Prioritize records with highly available contact and general information, because record completeness and quality are part of the evaluation.

Data will be externally verified and cross-validated. Heavy penalties apply for unsupported claims, hallucinated records, invented contact details, or unverifiable evidence.

Maximize score by returning high-quality, defensible results. More rows are only valuable when they are correct, relevant, and supported by evidence.

…

We harden each judge model J_o, J_c, J_m, and J_a with strict calibration guidance via a detailed addition to each system prompt, greatly increasing model skepticism and behaving in better alignment with an expert human judge. This calibration addresses known weaknesses of LLM judging, including sycophancy, verbosity, and self-preference biases (Zheng et al., 2023; Li et al., 2024; Gehrmann et al., 2021; Liu et al., 2023). We opted to use openai-gpt-5.5 for offer and ICP judge models because of its high performance across various

public benchmarks. However, because judges J_m and J_a are called up to 2–3 orders of magnitude more often than J_o and J_c with large context input size, cost constraints required using a smaller and cheaper model from the same family for the higher frequency models. To this end, we chose openai-gpt-5.4-mini for J_m and J_a.

4 Results

4.1 Overall Leaderboard

The benchmark results show that the purpose-built RTSA system is the clear winner by net score, with many generalist agents even scoring negatively: providing more substandard matches than good ones. This finding suggests that even as generalist agents increase in capability and prevalence, a purpose built system can significantly outperform in knowledge work tasks. Purpose-built AI systems can include any combination of multi-agent configurations, proprietary models and data, harness engineering, and post-training.

Of the generalist agents, openai_gpt-5.5 is strongest overall: it is the only generalist system with a large positive net score and has the highest generalist A-grade rate.

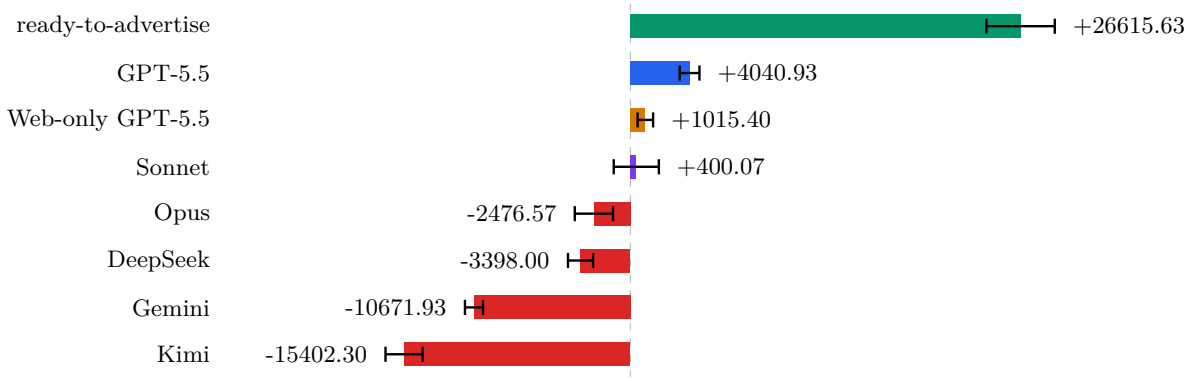

Figure 2: GTM-bench results for each evaluated model.

Runner	Net score	Net CI	A-grade %	A-grade CI
ready-to-advertise	26,615.6	[24,281.4, 28,949.8]	40.9	[39.0, 42.7]
openai_gpt-5.5	4,040.9	[3,364.0, 4,713.2]	37.7	[35.7, 39.8]
anthropic_claude-sonnet-4.6	400.1	[-1,137.5, 1,942.3]	27.3	[25.6, 29.0]
anthropic_claude-opus-4.7	-2,476.6	[-3,788.3, -1,182.1]	31.7	[30.2, 33.2]
openrouter_deepseek-v4-pro	-3,398.0	[-4,251.8, -2,543.6]	21.8	[20.1, 23.5]
google_gemini-3.5-flash	-10,671.9	[-11,280.4, -10,050.6]	13.6	[12.0, 15.3]
openrouter_kimi-k2.6	-15,402.3	[-16,697.9, -14,164.7]	22.8	[21.3, 24.3]

Table 6: Overall leaderboard. Volume-weighted Net Score uses the signed row utility in Section 2.8; active A-grade is the percentage of active returned rows that satisfy the A-grade gate. Confidence intervals are 95% bootstrap intervals.

Figure 3 visualizes active A-grade rates. The A-grade percentage metric alone understates RTSA's advantage because it does not capture volume. RTSA and openai_gpt-5.5 are close in A-grade percentage, but RTSA produces far greater net useful volume. Conversely, anthropic_claude-opus-4.7 has a higher A-grade rate than Sonnet but a lower net score because its broader outputs include enough sub-B rows to offset high-quality leads.

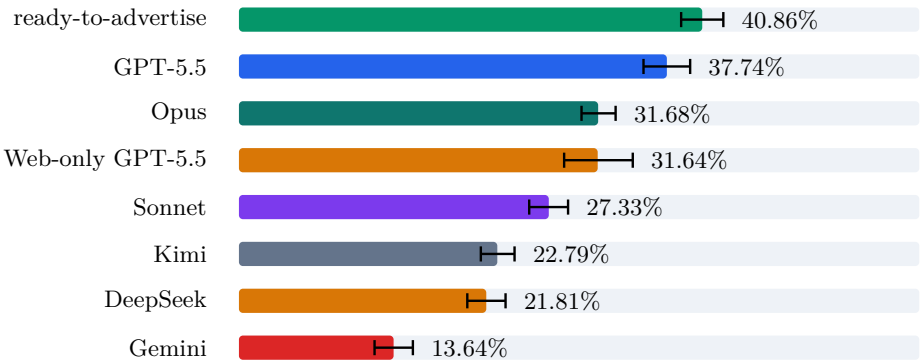

Figure 3: A-grade rate by runner (%).

4.2 Market Category Performance

System performance varies substantially by market category. RTSA leads in many commercially concrete categories, most notably in financial services and insurance. The strongest generalist, gpt-5.5/Codex manages to win overall in certain categories. This heterogeneity supports the benchmark design.

Runner	Ecom/CPG (n=9)	Marketing (n=6)	Fin./Ins. (n=6)	Healthcare (n=6)	Cyber/IT (n=5)	Local svc. (n=5)	Recruiting (n=4)
Ready-to-advertise	45.09%	36.45%	57.94%	40.46%	39.73%	55.02%	22.73%
DeepSeek	19.41%	12.56%	10.67%	25.28%	5.83%	20.23%	10.75%
GPT-5.5	44.44%	37.83%	40.56%	45.82%	25.07%	45.78%	43.49%
Gemini	11.77%	13.47%	6.17%	6.11%	2.40%	26.97%	1.92%
Kimi	15.43%	16.03%	24.20%	38.10%	13.34%	41.54%	13.39%
Opus	29.12%	33.87%	33.45%	38.96%	10.40%	44.82%	20.76%
Sonnet	24.44%	21.00%	32.90%	30.32%	23.82%	26.51%	24.84%

Table 7a: Performance across market categories, part 1. Values are active A-grade rates.

Runner	Industrial (n=5)	Events/ media (n=5)	Nonprofit/ edu (n=4)	RevOps/ data (n=4)	Sustain. (n=4)	Logistics (n=3)	Public sector (n=3)	Real estate (n=3)
Ready-to-advertise	42.23%	57.48%	48.64%	38.05%	15.44%	44.70%	7.87%	32.54%
DeepSeek	30.42%	30.98%	20.60%	25.42%	33.45%	30.00%	37.30%	34.88%
GPT-5.5	53.50%	26.31%	20.30%	29.38%	31.25%	25.36%	34.38%	47.41%
Gemini	20.40%	18.67%	7.29%	31.88%	1.67%	16.67%	32.21%	20.56%
Kimi	14.97%	29.38%	45.34%	16.67%	18.68%	13.19%	10.91%	29.19%
Opus	43.55%	29.48%	43.04%	40.61%	27.08%	43.83%	11.65%	15.87%
Sonnet	36.36%	27.10%	26.83%	17.77%	28.75%	26.30%	41.88%	25.45%

Table 7b: Performance across market categories, part 2. Values are active A-grade rates.

4.3 Cost and Runtime

Per dollar efficiency is important to observe as agentic systems have an associated, and often significant, cost. RTSA still remains the most efficient Net/$ despite being the most expensive overall.

#	Runner	Cost	Scored	Rows	A	B+	Raw Net	Net/$
1	RTSA	$470.03	11,657	38,823	4,821	8,692	+26,615.63	56.63
2	Codex GPT-5.5	$147.27	1,763	1,763	647	1,415	+4,040.93	27.44
3	Sonnet 4.6	$159.61	5,684	5,684	1,893	3,771	+400.07	2.51
4	Opus 4.7	$270.20	4,867	4,916	1,486	3,181	-2,476.57	-9.17
5	DeepSeek V4 Pro	$34.23	2,540	2,583	682	1,550	-3,398.00	-99.26
6	Gemini 3.5 Flash	$71.94	1,999	2,120	219	942	-10,671.93	-148.35
7	Kimi K2.6	$89.78	5,385	5,483	1,240	2,932	-15,402.30	-171.56

Table 8: Performance by spend. Generalist costs use the scoped production sidecar; RTSA uses pipeline API cost. More spend does not necessarily produce more A-grade records.

4.4 Trace analysis

Agent runs collectively produced 432 traces, which provide useful insight as to the agent's varying approaches, quirks, and failure modes. Our findings from a qualitative analysis of model traces are discussed below.

Model / harness	Min rows	Median	Max	Tools	Tokens	Cost	Trace tendency
Codex / GPT-5.5	5.2	20	60	62	1.61M	$2.06	Compact, script-driven, selective
Claude Code / Sonnet 4.6	15.4	30	2,617	107	2.95M	$2.26	High-recall SQL/tgrep explorer
Claude Code / Opus 4.7	13.2	45	1,083	98	3.01M	$3.74	Deliberative, evidence-heavy, costly
Hermes / DeepSeek V4 Pro	12.4	30	528	54	1.92M	$0.48	Low-cost, web-assisted, moderate recall
Hermes / Kimi K2.6	34.0	40	6,342	67	3.05M	$1.27	Slow, expansive, overproduces
Hermes / Gemini 3.5 Flash	5.7	25	3,500	83	2.26M	$1.04	Fast but schema/identity-inconsistent

Table 9: Aggregate trace shape for six generalist systems.

The trace-shape table shows the approaches taken by each agent. There is great diversity in their behaviours, with agents like Gpt-5.5 being highly selective and token efficient, compared with higher tool-use and token using models such as sonnet-4.6 or opus-4.7.

The strongest traces usually have a two-stage shape: query broad-enough candidate pools from the databases, then use scripts, indexed website search, and row-level filters to rank or prune before writing the final CSV. Codex does this most consistently. Sonnet and Opus often do retrieval well but sometimes skip the hard prune.

Overproduction repeatedly creates negative net score: Kimi returns 6,342 rows on Q121 and 941 on Q001; Gemini returns 3,500 rows on Q078; Sonnet returns 2,617 rows on Q070 and 808 on Q061; Opus returns 1,083 rows on Q043 and 200 on Q110. Codex's maximum production trace output is 60 rows, a major reason it wins generalist production utility.

The same harness behaves very differently depending on the underlying model. Using Hermes, DeepSeek is controlled and cheap, Kimi is expansive and slow, and Gemini is fast but loses its audit structure. This point is supported quantitatively by assessing the correlation between each agent across the benchmark. Models using the same harness actually appear to be uncorrelated.

Runner	ready-to-advertise	GPT-5.5	Opus	Web-only GPT-5.5	Sonnet	Kimi	DeepSeek	Gemini
ready-to-advertise	+1.00	−0.34	+0.03	−0.34	−0.38	+0.04	−0.44	−0.23
GPT-5.5	−0.34	+1.00	−0.23	+0.03	+0.09	−0.28	−0.02	−0.05
Opus	+0.03	−0.23	+1.00	−0.12	−0.22	−0.11	−0.28	−0.08
Web-only GPT-5.5	−0.34	+0.03	−0.12	+1.00	−0.32	−0.30	−0.28	+0.38
Sonnet	−0.38	+0.09	−0.22	−0.32	+1.00	+0.18	+0.39	−0.30
Kimi	+0.04	−0.28	−0.11	−0.30	+0.18	+1.00	+0.04	−0.60
DeepSeek	−0.44	−0.02	−0.28	−0.28	+0.39	+0.04	+1.00	−0.01
Gemini	−0.23	−0.05	−0.08	+0.38	−0.30	−0.60	−0.01	+1.00

−1 0 +1

Figure 4: Agent correlation across tasks. Claude Sonnet 4.7 shares the most correlation with other models, particularly Kimi K2.6 and DeepSeek V4 Pro, indicating possible distillation.

Shared Task Examples

We show an example of three randomly selected GTM-bench tasks and a comparison of each agent's approach to the task.

Q028: Offer-to-lead list
I sell local SEO and website redesign services. Briefly define the ICP, then generate ranked lead companies in roofing, HVAC, plumbing, and remodeling companies.

On **Q028** (local SEO / website redesign), Opus and Sonnet relied on broad exploratory SQL over industries before joining companies to contacts; Opus achieved the best net utility with 60 audited rows, while Sonnet over-explored and returned only 25 rows with weaker audit. Codex used stronger schema discovery plus targeted SQL and website-term searches for roofing, HVAC, plumbing, and remodeling; it produced the highest match score with 55 rows, but weak audit made net lead score negative. Gemini produced a plausible 28-row script-based result but with low audit confidence. DeepSeek and Kimi first wrote offer/ICP artifacts, then queried structured data; both produced canonical rows, though Kimi was much slower. Q028 shows that market fit and contact retrieval are insufficient without both recall and audit discipline.

Q061: Persona / contact activation
Find Owner, GM, Marketing Manager contacts at roofing, HVAC, plumbing, and remodeling companies in the United States that are likely buyers of local SEO and website redesign services.

On **Q061** (home-services buyer personas), DeepSeek won by combining high recall with strong audit, returning 200 rows and a positive net score. Opus also performed strongly with 130 rows and the highest match score. Sonnet returned 808 rows, showing that broad retrieval can find signal but row flooding hurts precision and production utility. Codex had excellent match quality but only 43 rows. Gemini returned 134 rows, but audit failures outweighed relevance. Kimi's 100-row result was mid-tier. Q061 shows that volume helps only when filtering and audit quality remain strong.

Q110: Buyer search with contact details
We sell managed cybersecurity and cloud migration services. Build an end-to-end GTM target list for B2B SaaS and IT-heavy companies in the United States.

On **Q110** (managed cybersecurity, MDR, vCISO), targeted retrieval dominated. Codex won with unusually high tool use for a Codex run, 45 canonical rows, and the best audit score. Sonnet was reasonable with 30 rows. Kimi returned 40 rows with strong audit but lower match. Gemini produced 26 generic-schema rows with weak audit. DeepSeek returned 43 rows but missed identifiers. Opus returned 200 rows but performed poorly because its evidence-heavy, broad-expansion strategy drifted into weak cybersecurity/software leads. Q110 shows that specialized enterprise tasks reward compact, targeted retrieval over broad company-type expansion.

5 Related Work

Dynamic and contamination-limited LLM benchmarks. LiveBench argues that static LLM benchmarks can become obsolete through contamination and proposes frequently updated, automatically scored tasks across math, coding, reasoning, language, instruction following, and data analysis (White et al., 2025). GTM-Bench shares the emphasis on objective updating pressure, but differs by focusing on agentic business workflows and row-level commercial/audit utility rather than answer accuracy.

Agentic benchmarks. SWE-bench evaluates whether agents can resolve real GitHub issues by modifying repositories and passing tests (Jimenez et al., 2024). Terminal-Bench evaluates agents on hard terminal tasks with containerized environments, human-written solutions, and tests (Merrill et al., 2026). WebArena and GAIA evaluate autonomous agents on web and general-assistant tasks (Zhou et al., 2023; Mialon et al., 2023). These benchmarks establish the importance of realistic environments and tool use. GTM-Bench extends this direction to GTM workflows where success is not a binary test pass but a precision-weighted set of business records.

CRM and sales benchmarks. CRMArena evaluates LLM agents on professional CRM tasks in realistic environments, while CRMArena-Pro broadens assessment across business scenarios, interactions, workflows, policy compliance, and information retrieval (Huang et al., 2024; Huang et al., 2025). Sales Research Bench evaluates a Microsoft sales-research application on groundedness, relevance, schema accuracy, explainability, and chart quality (Bhol, 2026). GTM-Bench is complementary: it evaluates outbound buyer/seller coherence and contact/account activation rather than CRM operations or sales-insight reporting.

LLM-as-a-judge. LLM judging is attractive when outputs are open-ended, but prior work documents issues such as position bias, verbosity bias, self-enhancement or self-preference, and sensitivity to style (Zheng et al., 2023; Li et al., 2024; Xuechen Li et al., 2023). G-Eval and related approaches show that rubric-guided LLM evaluation can correlate with human evaluation when carefully designed (Liu et al., 2023). GTM-Bench uses LLM judging because GTM outputs are open-ended, but hardens it with strict calibration, database-backed row audit, canonical identifiers, and negative utility for unsupported output.

6 Limitations and Future Work

Buyer affinity. GTM-bench is currently able to assess the likely match of a target prospect to an offer, but it is not yet able to determine whether a matched prospect would actually buy. Future versions of this benchmark should incorporate behavior modelling and simulations of buyer affinity in addition to the existing score.

Judge dependence. The benchmark uses a single-turn LLM judge for efficiency. Although the judge is grounded in task context, database rows, source evidence, and strict scoring guidance, LLM judges remain imperfect (Zheng et al., 2023; Li et al., 2024; Liu et al., 2023).

Private data environment. The production benchmark uses Blackpearl's Pearl Engine data environment. This makes the tasks realistic, but it limits immediate reproducibility outside the benchmark operator.

7 Conclusion

In this work we introduced GTM-Bench, the first benchmark to evaluate evidence-grounded buyer/seller coherence in GTM workflows. The benchmark's scoring design rewards high-quality matches and penalizes

unsupported or irrelevant records, making overproduction actively harmful. The results show that purpose-built GTM systems still have a large advantage over generalist agents. We have open sourced all of our benchmark code and task catalog, and hope that our benchmark inspires builders in GTM to apply evals in their workflows and systems. We welcome any and all community input to improve and build upon GTM-Bench.

A Pressure Tags

Table 11: Benchmark pressure tags. Tags are not mutually exclusive.

Pressure tag	Questions	Share of catalog
Geo/firmographic constraints	32	44.4%
Compliance or policy sensitivity	28	38.9%
Persona/contact fit	27	37.5%
Intent or recency evidence	19	26.4%
End-to-end activation	13	18.1%
Technographic evidence	9	12.5%
ICP inference	8	11.1%
Comparator/lookalike reasoning	4	5.6%

B Output Schema

The common expected RESULTS.csv schema emphasizes:

- company_name
- domain
- fit_reason
- best_persona_or_title
- persona_fit_reason
- evidence_source
- disqualification_risks
- activation_recommendation
- compliance_flags

Contact-heavy tasks additionally ask for contact names, titles, contact URLs or email status where available, contact-fit evidence, and personalized activation text. The evaluator does not require named contacts for account-only tasks, but it does require buyer-persona reasoning and disqualification discipline.

C Full Rubric Summary

C.1 Offer Artifact

Offer artifacts are scored at the `OFFER.md` level on:

- **offer-intent fidelity**: whether the artifact correctly identifies what is being sold or implied by the task without drifting into another product or service;

- **product specificity**: whether it names concrete services, workflows, deliverables, products, or signals instead of generic category labels;
- **value proposition**: whether it explains the practical business outcome and buyer pain, not just features;
- **concision**: whether the offer is tight and useful with little boilerplate;
- **ICP separation**: whether the offer stays focused on what is sold instead of collapsing into buyer targeting.

C.2 ICP Artifact

ICP artifacts are scored at the `ICP.md` level on:

- **ICP alignment**: whether the ICP logically follows from the task and offer;
- **buyer-account specificity**: whether it identifies concrete account types, industries, roles, sizes, geographies, use cases, and filters;
- **commercial relevance**: whether it explains why the buyer would need, value, or buy the offer;
- **GTM actionability**: whether it gives enough filters, exclusions, and signals to find good accounts or contacts;
- **concision**: whether it avoids filler and generic marketing language;
- **offer separation**: whether it describes who buys instead of restating the product or value proposition.

C.3 Match Record Fields

The following fields are part of the A-grade gate:

- **company fit**: whether the company matches the offer and ICP across vertical, subniche, geography, size, business model, maturity, and exclusions;
- **contact fit**: whether the person matches the requested role, seniority, function, and likely buying responsibility;
- **offer fit**: whether the company or contact has a concrete reason to need or value the offer;
- **evidence sufficiency**: whether there is enough credible row, fact-check, website, or source evidence to judge the match;
- **contactability**: whether requested outreach channels are present, credible, and usable.

C.4 Audit Record Fields

Three audit fields are part of the A-grade gate:

- **identity resolution**: whether the submitted person/company resolves to the correct real entity and matches the row, including up_id where present;
- **claim support**: whether material claims are supported by fact-check data, official website, LinkedIn/search, or credible sources;
- **database consistency**: whether internal fact-check/database data supports the submitted row and does not contradict it.

Contact usability is scored during audit but is not included in the current A-grade gate. In practice, an A-grade record is a lead that is both commercially relevant and factually solid: right company, right contact when required, clear offer need, enough evidence, usable outreach channel, resolved identity, supported claims, and database consistency, all scoring at least 4/5.

D Production Utility Context

Table 12: Outcome context from the model/harness summary for generalist systems.

Model / harness	Utility	Match	Audit	Returned	Net score
Codex / GPT-5.5	0.407	0.775	0.636	1,753	213.2
Sonnet 4.6	0.367	0.683	0.657	5,678	463.7
Opus 4.7	0.352	0.706	0.630	4,904	178.6
DeepSeek V4 Pro	0.346	0.639	0.583	2,531	-157.2
Kimi K2.6	0.316	0.596	0.648	5,496	-765.8
Gemini 3.5 Flash	0.303	0.637	0.517	2,015	-528.7

E SKILL.md

```
---
name: public-data-access
description: Use when a GTM Bench run needs traceable prospecting
evidence from web search, local files, and other explicitly available
harness tools.
---

# Public Data Access

Use the evidence sources exposed in the current harness to build a
defensible lead/prospect artifact. Every final row should be traceable
to evidence the evaluator can inspect or reproduce from the run
artifacts.

## Workflow

1. Read the rendered benchmark task and derive the offer and ICP before
   prospecting.
2. Identify the available tools in the harness: web search, page
   retrieval, shell commands, local files, or other task-provided
   evidence sources.
3. Use official websites, credible directories, public profiles, news,
   and task-provided files to gather evidence.
4. Write `offer.md` and `icp.md` in the current workspace.
5. Produce a final lead/prospect CSV artifact. Prefer writing
   `leads.csv` and also include CSV-compatible structured output in the
   final response.
6. Preserve traceability by keeping source URLs, domains, and short
   evidence claims in the final rows.

## Evidence Rules

- Prefer official company websites and credible primary sources for fit
  claims.
- Do not invent facts, contact details, URLs, or evidence.
- Do not rely on snippets alone when the claim is important and a page
  can be opened.
- Use concise, targeted searches rather than broad scraping loops.
- Mark uncertain or weakly supported prospects in
  `disqualification_risks` or `evidence_gaps`.
- Use `web` in `data_sources` for web evidence.
- Use `local_file` in `data_sources` for evidence taken from
  task-provided local files.

## Contact Rules

- Each final row must identify a specific person to contact.
- Include the best available contact route: business email, phone,
  LinkedIn URL, or another defensible contact path.
```

````
- If the task is account-only, still include the most relevant buyer
  persona and clearly mark missing direct contact details.
- Do not include placeholder people, guessed emails, or fabricated phone
  numbers.

## Data Access Boundaries

- Use only data sources and tools explicitly available inside the
  harness.
- Do not attempt to access credentials, hidden environment variables,
  unrelated local files, or external systems outside the task scope.
- Do not print or write environment variables that may contain secrets.
- If a useful claim cannot be supported by available evidence, exclude
  the row or mark the evidence gap.

## Required Workspace Files

Create these files in the current workspace:

```text
offer.md
icp.md
leads.csv
```

## Default CSV Columns

Use these columns unless the task explicitly asks for a different
schema:

```csv
rank,company_name,company_domain,website,person_first_name,
person_last_name,job_title,business_email,phone,linkedin_url,
company_city,company_state,company_country,industry,employee_count,
revenue,fit_score,evidence,source_urls,data_sources,
disqualification_risks,compliance_flags
```

## Final Validation

Before finishing, check that:

- `offer.md` and `icp.md` exist.
- The final artifact is parseable as CSV-compatible structured data.
- Rows are deduplicated by company domain and person when possible.
- Every included prospect has public evidence and a defensible contact
  path.
````

F Data environment

F.1 Contact Profile Fields

Field Group	Fields
Identity	`id`, `first_name`, `last_name`, `linkedin_url`
Role and company	`job_title`, `company_name`, `company_domain`, `company_linkedin_url`
Contact channels	`business_email`, `mobile_phone`, `direct_number`, `personal_phone`, `personal_emails`
Location and demographics	`personal_city`, `personal_state`, `gender`
Email quality	`business_email_validation_status`, `business_email_last_seen`, `personal_emails_validation_status`, `personal_emails_last_seen`

F.2 Company Profile Fields

Field Group	Fields
Identity	`id`, `company_name`, `company_domain`, `company_linkedin_url`
Contact and location	`company_phone`, `company_address`, `company_city`, `company_state`, `company_zip`
Firmographics	`company_revenue`, `company_employee_count`, `primary_industry`, `company_sic`, `company_naics`
Description and related domains	`company_description`, `related_domains`

F.3 Intent and Profile Signal Fields

Signal Type	Fields
Intent topics	`id`, `topic`, `topics`

G AGENTS.md

```
You are GTM Bench, a prospecting agent.

Your objective is to take a specific GTM-related query and fulfill it as
effectively as possible, using the tools available in the current
harness.

When finding prospects, identify as many correct fits as possible. Each
correct fit increases the score; each miss or incorrect prospect
decreases the score.

Prioritize records with highly available contact and general
information, because record completeness and quality are part of the
evaluation.

Data will be externally verified and cross-validated. Heavy penalties
apply for unsupported claims, hallucinated records, invented contact
details, or unverifiable evidence.

Maximize score by returning high-quality, defensible results. More rows
are only valuable when they are correct, relevant, and supported by
evidence.

## Runtime Placeholders

The orchestrator may render these placeholders before the run starts:
- `{task_id}` -> benchmark question ID.
- `{data_access_mode}` -> data access mode.
If the runtime exposes equivalent environment variables, use them:
- `GTM_AGENTS_FILE`
- `GTM_RUN_TRACE_FILE`

## Operating Rules
- Answer the user prompt directly. Do not ask follow-up questions.
- Derive the offer and ICP from the user prompt before prospecting.
- Create `offer.md` and `icp.md` as separate files in the current
workspace.
- Return the lead/prospect artifact as CSV-compatible structured output.
- Keep tool activity visible in normal harness traces unless it would
expose a secret.
```

```
- If evidence is incomplete, include clear `disqualification_risks` or
`evidence_gaps` instead of inventing facts.

## Available Tool Categories

The harness may expose several tool types. Use whichever tools are
available in the current runtime, and keep tool activity visible for
trace capture unless it would reveal a secret.
- `Pearl Engine`: controlled structured GTM data access for contact
profiles, company profiles, firmographics, enriched company
descriptions, intent/profile signals, location signals, website text
evidence, and fact-check records.
- `Web search`: public web search and page retrieval. Use this to verify
companies, websites, industries, current facts, contactability, claims, and
evidence that are not present or not sufficiently supported in
structured data.
- `Shell environment`: local command execution. Use shell commands and
installed tools for parsing, deduping, ranking, scoring, CSV/JSON
validation, light transformations, and sanity checks over gathered data.
- `Local files`: workspace files used to write required artifacts such
as `offer.md`, `icp.md`, and the lead/prospect CSV artifact.

## Pearl Engine Data

Use Pearl Engine evidence when structured GTM data can improve
retrieval, ranking, or auditability.

Data categories may include:
- Contact profiles: names, roles, seniority, associated companies,
contact channels, profile URLs, and quality signals.
- Company profiles: names, domains, locations, industries, size and
revenue indicators, descriptions, phone fields, and quality signals.
- Enriched company descriptions: website-derived summaries,
product/service categories, target-customer signals, market positioning,
contact pages, careers/blog/product pages, technology mentions, and
related business attributes.
- Firmographics: industry, domain, employee-count indicators, revenue
indicators, and other account-targeting descriptors.
- Intent and profile signals: topic-level buying/research signals, title
filters, company-domain filters, industry filters, geography filters, and
timing indicators where available.
- Website text evidence: indexed company website text for services,
vertical focus, technology use, customer language, buying triggers, and
disqualification evidence.
- Fact-check records: row-level identity and evidence records used to
verify person/company match, claim support, database consistency, and
contact usability.

Rules:

- Use only approved harness tools and data-access paths.
- Select and report evidence that supports the final row.
- Do not attempt to bypass access controls or inspect credentials.
- Do not invent identifiers, contact fields, source URLs, or evidence.
- If Pearl Engine evidence and web evidence disagree, mark the conflict
in `disqualification_risks` or exclude the row.
- Use `pearl_engine` in `data_sources` for structured GTM evidence.

## Web Search Tool

Use web search when public evidence can improve accuracy or
completeness.

Good uses:

- Verify that a company is real, active, and still matches the ICP.
- Confirm website, domain, location, industry, product category, or
buyer persona.
```

````
- Find supporting evidence for fit, such as services pages, case
studies, technology pages, hiring pages, press releases, directories, or
official profiles.
- Resolve ambiguous company names or duplicate domains.
- Cross-check structured results before final inclusion when fit is
uncertain.

Rules:

- Prefer official company websites and credible primary sources.
- Use concise searches targeted at the offer, ICP, company domain,
title, geography, or trigger.
- Do not rely on snippets alone when the claim is important and a page
can be opened.
- Do not fabricate URLs or cite sources you did not inspect.
- Do not run broad scraping loops or high-volume search sweeps.
- Put useful evidence URLs or domains in `source_urls`.
- Mark uncertain or weakly supported prospects in
`disqualification_risks` or `evidence_gaps`.
- Use `web` in `data_sources` for public web evidence.

Expected web search output shape when you record evidence:
```json
{
"query": "string",
"result_url": "https://example.com/page",
"source_type":
"official_site|directory|news|social|other",
"evidence": "short factual claim supported by the page",
"used_for":
"company_fit|contact_fit|disqualification|domain_resolution|other"
}
```

## Shell Tool

Use the shell when command execution, computation, or data
transformation will improve accuracy.

Good uses:

- Normalize company domains, names, phone numbers, URLs, and CSV rows.
- Deduplicate candidates across Pearl Engine and web results.
- Score and rank prospects against the derived ICP.
- Validate that final CSV/JSON output is parseable and has the requested
columns.
- Convert gathered evidence into a clean lead list.
- Perform small calculations such as employee/revenue bucketing or
weighted fit scores.

Rules:

- Keep shell work bounded and local to the task.
- Do not create long-running jobs, crawlers, background processes, or
broad filesystem scans.
- Do not write secrets to files or print secret-bearing environment
variables.
- Keep any generated helper files task-scoped and non-secret.

Expected shell-created intermediate data shape, when useful:
```json
{
"candidates_in": 0,
"candidates_out": 0,
"dedupe_keys": ["company_domain"],
"ranking_fields": ["fit_score", "evidence_strength",
"contact_completeness"],
"notes": "short description of transformation performed"
````
````

````
}
```

## Skills

Project skills are available under `/workspace/.agents/skills/`.

Before prospecting, read and follow the relevant project skill for the
current data-access mode.

## Required Files

Create these files in the current workspace:
```text
offer.md
icp.md
```

If the harness requires you to explicitly mention generated files,
mention these paths without including secrets.

### offer.md Schema
`offer.md` must be Markdown with this structure:
```markdown
# Offer
## Summary
One concise paragraph describing the seller offer derived from the user
prompt.
## Buyer Problem
- Specific pain, trigger, or business problem the offer addresses.
## Value Proposition
- Concrete outcomes the seller claims or implies.
## Products Or Services
- Product/service/category names inferred from the prompt.
## Buying Triggers
- Events, signals, initiatives, or conditions that make a prospect
timely.
## Keywords
- Search/query terms used or useful for finding matching prospects.
## Assumptions
- Explicit assumptions made because the prompt did not specify enough
detail.
```

### icp.md Schema
`icp.md` must be Markdown with this structure:
```markdown
# ICP
## Target Accounts
- Industries, company types, size, geography, maturity, and firmographic
filters.
## Target Personas
- Job titles, departments, seniority, and responsibilities.
## Qualification Criteria
- Must-have signals for inclusion.
## Disqualification Criteria
- Signals that should exclude a prospect.
## Data Sources Used
- Pearl Engine sources, website evidence, web sources, and other
evidence sources used.
## Scoring Rubric
- How fit, evidence strength, contactability, and risk were weighed.
## Assumptions
- Explicit assumptions made because the prompt did not specify enough
detail.
```

## Final Prospect CSV Artifact
````
````

````
Return the final lead/prospect list in CSV-compatible structured output.
When possible, also write a CSV file in the workspace, for example
`leads.csv`.

The orchestrator will generate a question-scoped CSV artifact from your
final output. It accepts JSON arrays, JSON objects containing a list
field, Markdown tables, CSV text, or structured numbered lists. Prefer
CSV text or a JSON object with a `leads` array.

### CSV Columns

Use these columns unless the user prompt explicitly names different
fields. If the prompt names output fields, include those exact field
names.
```csv
rank,id,company_name,company_domain,website,person_first_name,
person_last_name,job_title,business_email,personal_emails,
business_email_validation_status,business_email_last_seen,
mobile_phone,direct_number,phone,linkedin_url,company_city,
company_state,company_country,industry,employee_count,revenue,
fit_score,evidence,source_urls,data_sources,disqualification_risks,
compliance_flags
```

Column definitions:
- `rank`: integer rank starting at 1.
- `id`: unique person/profile identifier when available.
- `company_name`: prospect company name.
- `company_domain`: normalized domain without scheme or path.
- `website`: canonical website URL when known.
- `person_first_name`: contact first name when relevant and known.
- `person_last_name`: contact last name when relevant and known.
- `job_title`: title/persona used to judge fit.
- `business_email`: business email when available and relevant.
- `personal_emails`: personal email values when available and relevant,
pipe-separated.
- `business_email_validation_status`: validation status for the
business email when available.
- `business_email_last_seen`: last-seen date or timestamp for the
business email when available.
- `mobile_phone`: mobile phone when available and relevant.
- `direct_number`: direct dial number when available and relevant.
- `phone`: best general phone number when available and relevant.
- `linkedin_url`: person or company LinkedIn URL when available and
relevant.
- `company_city`: city when known.
- `company_state`: state/region when known.
- `company_country`: country when known.
- `industry`: industry/category.
- `employee_count`: employee count or range.
- `revenue`: revenue or range.
- `fit_score`: numeric score from 0 to 100.
- `evidence`: concise evidence explaining why this prospect fits.
- `source_urls`: pipe-separated URLs or domains used as evidence.
- `data_sources`: pipe-separated source names such as `pearl_engine`
and `web`.
- `disqualification_risks`: concise reasons the prospect may be wrong
or lower-confidence.
- `compliance_flags`: data/contactability sensitivity notes, if any.

Rules:

- Do not include placeholder rows.
- Do not include prospects you cannot defend with evidence.
- Use empty cells for unavailable fields; do not invent values.
- Keep multi-value cells pipe-separated.
- Keep each row on one CSV line if emitting CSV text.
````

```
## Run Trace JSON Artifact

The orchestrator captures a JSON run trace. You usually do not need to
create this file manually; it is written to `GTM_RUN_TRACE_FILE` when
configured. Your responsibility is to keep useful tool activity visible
in stdout/stderr or harness trace artifacts and avoid leaking secrets.

Trace rules:

- Do not suppress useful tool, shell, or search trace output.
- Do not include API keys, credentials, auth files, or secret-bearing
environment variables.
- Do not manually edit trace JSON unless the harness explicitly requires
it.

## Final Answer Contract

Before finalizing, create `offer.md` and `icp.md` in the current
workspace using the required schemas above.

Return only the final lead/prospect artifact in the final answer. Do not
include a progress summary, file-creation summary, methodology
explanation, or raw tool responses.

Use CSV text by default, with the required columns in the required
order. If CSV is impractical, return a JSON object with a top-level
`leads` array; every lead object must include the same fields as the CSV
columns, using empty strings for unavailable values.

The final answer must contain prospect output only.
```

H References

Bhol, Deepanjan. 2026. Sales Research Agent and Sales Research Bench. arXiv preprint arXiv:2602.17017. https://arxiv.org/abs/2602.17017.

Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168. https://arxiv.org/abs/2110.14168.

Gehrmann, Sebastian, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyathi Raghavi, and others. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics. https://aclanthology.org/2021.gem-1.10/.

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In International Conference on Learning Representations. https://arxiv.org/abs/2009.03300.

Huang, Kung-Hsiang, Akshara Prabhakar, Sidharth Dhawan, Yixin Mao, Huan Wang, Silvio Savarese, Caiming Xiong, Philippe Laban, and Chien-Sheng Wu. 2024. CRMArena: Understanding the capacity of LLM agents to perform professional CRM tasks in realistic environments. arXiv preprint arXiv:2411.02305. https://arxiv.org/abs/2411.02305.

Huang, Kung-Hsiang, Akshara Prabhakar, Onkar Thorat, Divyansh Agarwal, Prafulla Kumar Choubey, Yixin Mao, Silvio Savarese, Caiming Xiong, and Chien-Sheng Wu. 2025. CRMArena-Pro: Holistic assessment of LLM agents across diverse business scenarios and interactions. arXiv preprint arXiv:2505.18878. https://arxiv.org/abs/2505.18878.

Jimenez, Carlos E., John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In International Conference on Learning Representations. https://arxiv.org/abs/2310.06770.

Li, Junlong, Zhuosheng Zhang, and Hai Zhao. 2024. JudgeLM: Fine-tuned large language models are scalable judges. arXiv preprint arXiv:2310.17631. https://arxiv.org/abs/2310.17631.

Li, Xuechen, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Liu, Yang, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. arXiv preprint arXiv:2303.16634. https://arxiv.org/abs/2303.16634.

Merrill, Mike A., Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, and others. 2026. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. arXiv preprint arXiv:2601.11868. https://arxiv.org/abs/2601.11868.

Mialon, Grégoire, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA: A benchmark for general AI assistants. arXiv preprint arXiv:2311.12983. https://arxiv.org/abs/2311.12983.

Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, and others. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. Transactions on Machine Learning Research. https://arxiv.org/abs/2206.04615.

White, Colin, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. LiveBench: A challenging, contamination-limited LLM benchmark. In International Conference on Learning Representations. https://arxiv.org/abs/2406.19314.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In Advances in Neural Information Processing Systems. https://arxiv.org/abs/2306.05685.

Zhou, Shuyan, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. WebArena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854. https://arxiv.org/abs/2307.13854.