LLM-Based Agents for Competitive Landscape Mapping in Drug Asset Due Diligence

Alisa Vinogradova*, Vlad Vinogradov*, Dmitrii Radkevich, Ilya Yasny, Dmitry Kobyzev,
Ivan Izmailov, Katsiaryna Yanchanka, Roman Doronin, Andrey Doronichev

Bioptic.io, San Francisco, CA
info@optic.inc

Abstract

In this paper, we describe and benchmark a competitor-discovery component used within an agentic AI system for fast drug asset due diligence. A competitor-discovery AI agent, given an indication, retrieves all drugs comprising the competitive landscape of that indication and extracts canonical attributes for these drugs. The competitor definition is investor-specific, and data is paywalled/licensed, fragmented across registries, ontology-mismatched by indication, alias-heavy for drug names, multimodal, and rapidly changing. Although considered the best tool for this problem, the current LLM-based AI systems aren't capable of reliably retrieving all competing drug names, and there is no accepted public benchmark for this task. To address the lack of evaluation, we use LLM-based agents to transform five years of multimodal, unstructured diligence memos from a private biotech VC fund into a structured evaluation corpus mapping indications to competitor drugs with normalized attributes. We also introduce a competitor validating LLM-as-a-judge agent that filters out false positives from the list of predicted competitors to maximize precision and suppress hallucinations. On this benchmark, our competitor-discovery agent achieves 83% recall, exceeding OpenAI Deep Research (65%) and Perplexity Labs (60%). The system is deployed in production with enterprise users; in a case study with a biotech VC investment fund, analyst turnaround time dropped from 2.5 days to ~3 hours (~20x) for the competitive analysis.

Introduction

Mapping the entire competitive landscape for a drug is the first and important step of the competitive analysis underlying the patent, business, and scientific due diligence for in-/out-licensing, as well as for identifying gaps when planning a clinical trial for that specific asset. Regulators make competitor discovery increasingly important, since 12 January 2025, the EU Health Technology Assessment Regulation applies Joint Clinical Assessments (JCAs) (European Commission 2025) for new oncology medicines and all Advanced Therapy Medicinal Products (ATMPs). Missing a relevant competitor or proposing an inappropriate one can lead the JCA to set comparators that the drug developer did not anticipate, exposing the asset to European Union (EU)-wide market access and pricing risk; previously, such exposure was confined to national decisions Globally, expec-
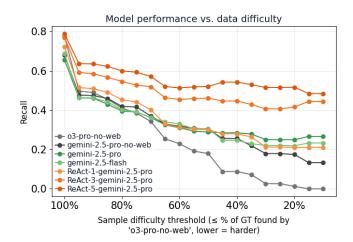
Figure 1: **Model performance across varying levels of sample difficulty.** The x-axis denotes difficulty thresholds: at each point, we evaluate all models on the subset of samples where *o3-pro-no-web* recovered ≤ the indicated percentage of ground-truth competitors. This allows us to assess how different agents perform on increasingly difficult samples, using a non-web baseline as the difficulty proxy.

tations align: ICH E10 (ICH 2000) and FDA effectiveness guidances (FDA 2023) emphasize appropriate control/comparator selection, and NICE (NICE 2022) methods require justifiable comparators in routine practice.

Identifying all competitors for an indication demands consolidating evidence from press releases, scientific literature, patents, clinical-trial registries, and target-space intelligence, then linking novel targets to the drugs that modulate them. Because information is scattered, alias-ridden, multilingual, and ranked differently by region and the landscape changes rapidly without central tracking, teams still rely on manual expert workflows and spreadsheets, creating a persistent completeness, recency, and cycle-time risks that cascade into comparator strategy, trial design, pricing, and licensing decisions.

Large language models (LLMs) appear well-suited to this task: they are increasingly multimodal and multilingual, can automate operating with scientific tools, web browsing, managing external databases at scale, and are being rapidly
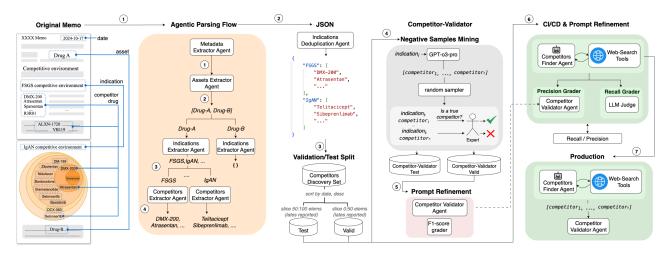
Figure 2: **Competitors Discovery System Diagram.** Diligence memos are parsed by a hierarchical multi-agent system into normalized JSON (assets, indications, competitors), deduplicated and split chronologically. Precision is graded by an expert-trained LLM-as-a-judge Competitor-Validator with mined negatives, recall by an LLM-as-a-judge that uses Web-search; both feed CI/CD. In production, a Competitors-Finder agent calls web tools and the Validator filters candidates.

operationalized across big pharma. For example, Moderna shifted from an internal GPT-4 API agent ("mChat") to ChatGPT Enterprise for thousands of employees, spawning 750+ internal GPTs including "Dose ID" (OpenAI 2024); Takeda deployed a secure Azure OpenAI assistant with PwC/Microsoft (PwC Germany 2024); and Bayer's ChatGPT-4 Turbo–based MyGenAssist cut pharmacovigilance letter time by 23% (Benaïche et al. 2025). Moreover, recent studies report that general-purpose, frontier LLMs can be competitive with—or even outperform—task-specific fine-tuned models on certain pharma-adjacent evaluations (Chen et al. 2025; McDuff et al. 2025). Accordingly, since marginal gains in this setting typically come from agentic scaffolds and execution policy rather than weight adjustments to a specific task, and apparent prompt wins can degrade under real-world constraints, it is necessary to prioritize end-to-end, domain-grounded benchmarking that exercises failure modes, to ensure deployed systems remain stable and maintain certain Service Level Agreements (SLAs).

There is no publicly accepted benchmark for competitor discovery because key evidence and curated lists are paywalled and heterogeneous across providers (e.g., Clarivate, Citeline, GlobalData, AlphaSense), necessitating in-house expert curation, and the definition of a "competitor" is investor- and stakeholder-specific. This task heavily relies on web browsing and multi-hop navigation across different sources, backtracking, and evidence reconciliation. Web-agent benchmarks such as BrowseComp (hard-to-surface web facts) (Wei et al. 2025), WebVoyager (interacting with real-world websites) (He et al. 2024), WebArena (realistic but closed-world sites) (Zhou et al. 2023), Mind2Web (generalist website tasks) (Deng et al. 2023), and WebLINX (conversational web navigation) (Lù, Kasner, and Reddy 2024) are not sufficient to capture the complexity of the competitors discovery problem: nomenclature harmonization; landscape completeness, long-tail coverage; and high-

recall retrieval at regulator-grade. To fill this gap, we use LLM-based agents to transform five years of multi-modal, unstructured diligence memos from a private biotech VC fund into a structured evaluation corpus that maps indications to competitor drugs with normalized attributes.

Because the primary deployment risk is incompleteness, our benchmark first makes recall observable — how many expert-identified competitors the agent recovers. At the same time, recent work presenting ChiDrug (Wu et al. 2025), a benchmark comprising six Chinese medication subtasks including an indication task, shows frontier models (e.g., GPT-4o, Claude 3.5) both omit valid answers (recall failures) and hallucinate (precision failures), underscoring the need to measure completeness and to validate outputs. We therefore introduce the post-retrieval filtering LLM-as-judge (Competitors Validator), which, in the fashion of the Paper-Bench's "expert slice" (Starace et al. 2025), uses a conservative synthetic-labeling step in which a calibrated agent, tuned to a 90% F1-score of detecting false-positive competitor drugs, proposes false candidates that are then filtered out before propagating them to the client-facing components of the system.

Another challenge in productionizing LLM applications is balancing the "bitter lesson" (that durable gains come from general methods that scale rather than hand-crafted domain rules) with the empirical need for agentic scaffolds on long-tail, multi-hop web tasks. As single-pass prompting, even with comprehensive Chain-of-Thought (CoT) prompting, reasoning language model, extensive tuning for tool use, often misses a large fraction of long-tail, multi-hop facts because the optimization landscape is too stiff to plan multi-step web navigation, branch or backtrack (Biran et al. 2024), (Gema et al. 2025). Agentic scaffolds that interleave reasoning and tool use, such as REACT (Yao et al. 2023) and REFLEXION (Shinn et al. 2023), alleviate this by decomposing the search space and learning from self-critique.

An example would be Grok-4 Heavy, which credits their performance jump from 44% to 50% on Humanity's Last Exam to parallel agentic search (xAI 2025). We therefore compare multiple model–scaffold configurations on the end-to-end task of enumerating all competitor drugs for a given indication and show that scaffolded agents reliably outperform single-pass prompting, especially on harder, fragmented cases.

Motivated by these gaps and the scaffold–vs.–"bitter lesson" trade-off, we build and evaluate a competitor-discovery agent and a post-retrieval LLM-as-judge (Competitor-Validator) on a new, domain-grounded benchmark. To our knowledge, no prior public benchmark evaluates competitor discovery; accordingly, we report baselines for OpenAI Deep Research and Perplexity Labs and find that our agent achieves 83% recall (vs. 65% and 60%, respectively) while the validator suppresses false positives to maximize precision. The agent is deployed as a component of a fast due-diligence system used by enterprise users; in a VC case study, analyst turnaround time fell from ∼2.5 days to ∼3 hours (∼20×).

Data

Essentially, we derive three datasets from the same memos corpus:

1. Competitors Dataset
Ground truth for enumerating competitors per indication (used to assess recall/coverage):

$$\mathcal{D} = \big\{ (ind_j, \mathcal{C}_j^\star) \big\}_{j=1}^M, \qquad \mathcal{C}_j^\star = \{d_{j1}, \ldots, d_{jK_j}\}.$$

Here, $\mathcal{C}_j^\star$ is expert-curated from private biotech VC fund memos and is *not assumed complete* and is subjective; it reflects what competitors the experts have identified in the past during their due diligence process on specific assets.

2. Attributes Dataset
Canonical drug attributes to assess agents ability to recover attributes for the identified competitors.

$$\mathcal{A} = \big\{ \big(d, \, ind^*, \, \mathbf{a}_d^{ind^*}\big) \big\}, \qquad ind^* \in \mathcal{I} \cup \{\varnothing\},$$

where $ind^* = \varnothing$ denotes that no indication is provided (i.e., the attribute is indication-independent); $\mathbf{a}_d^{ind^*}$ includes: drug name aliases, modality (type), mechanism(s) of action, targets, development stage, regulatory status, therapeutic area, other indication(s), administration routes, and company information (website/description/ticker).

3. Competitor-Validator Dataset
Pairs for tuning (prompt refinement) and applying the post-retrieval precision filter (LLM-as-judge):

$$\mathcal{V} = \big\{ (ind_i, \, drug_i, \, y_i) \big\}_{i=1}^N, \qquad y_i \in \{0, 1\}.$$

Positives ($y_i=1$) are expert-confirmed competitors; negatives ($y_i=0$) are hard near-misses (e.g., umbrella terms, out-of-scope indications, withdrawn/unrelated programs). $\mathcal{V}$ is used to tune the Competitor-Validator toward a high F1-score for both rejecting false positives and not missing relevant competing drug names.

Raw Data Description

We use due diligence memos of the private biotech VC fund (see "Original Memo" in Fig. 2 for a schematic view of memos we used). The initial corpus contained 210 unclassified memos spanning 15 years of the VC fund's work. Out of these we selected reports created from 2019–2025 that (i) perform due diligence of companies with at least one drug asset or (ii) assess the competitive landscape of a specific indication. This filtering yielded 73 reports with 174 *drug-asset – indication* pairs.

The memos vary in language and format: authors scatter competitive-landscape details throughout free-form text and embed every kind of artifact — low-resolution plots, full-page figures, standard tables, screenshots of slide decks, and other hard-to-parse images. To handle this heterogeneity, we built a parsing agent that reads both text and images, traverses the memos' intended hierarchical structure: company → asset → indication → competitor list—and systematically extracts competitor drugs for every mentioned indication (see "Agentic Parsing Flow" in Fig. 2).

Then, for each extracted drug, the Attributes Parsing Agents extract indication-dependent and indication-independent attributes.

Hierarchical Memos Parsing

Our multi-agent memo parser follows a hierarchical extraction workflow (see "Agentic Parsing Flow" in Fig. 2). First, a *Drug-Assets extractor* identifies every unique drug asset analyzed in the memo. Each asset is then passed, in parallel, to N *Indications extractors*, which extract all unique indications the given asset was analyzed against; each resulting indication is subsequently routed to a *Competitors extractor*, which extracts the competitor drugs mentioned for the corresponding drug–indication pair.

Then an *Attributes Parser* consumes each *(drug, indication)* node and emits: drug name aliases, modality (type), lead indication, other indications, and administration routes, mechanism(s) of action, targets, development stage, and regulatory status, therapeutic area, and company information (website/description/ticker).

Quality Controls & Implementation. The parser is based on Google's Gemini-2.5 Pro without web browsing. We use Gemini's caching for efficient processing across the multiple extraction stages. All four extractor agents rely on template-based, JSON-schema prompts that (i) demand verbatim evidence from the memo and (ii) reject outputs lacking an explicit mention, thereby preventing hallucination while ensuring structured, fact-only extraction. For non-English memos, the pipeline involves a translator agent, based on Gemini-2.5 Pro with web browsing, to translate indications and drug names while preserving life sciences terminology.

All outputs undergo schema-on-write validation with Pydantic. While some attributes are strict, fields like `modality`, `route of administration`, and `development stage` are controlled-vocabulary (lookup) columns seeded by part of the possible values (e.g., for `modality`, seeds are 'small molecule',

`'monoclonal antibody'`; for route of administration, `'oral'`, `'subcutaneous'`) but with open-world extension — novel memo values are admitted and normalized downstream. This yields a semi-structured, schema-constrained design that mixes closed enumerations with extensible controlled vocabularies.

Normalization and Alias Resolution

The parsed JSONs mirrors the original report hierarchy: *asset → indications → competitors → attributes* (see "JSON" in Fig. 2). For evaluation, we drop the top-level *asset* node and flatten to an indication-centric mapping: *indication → competitor drug → attributes*; where each drug links to its attributes.

Different asset memos can refer to the same indication — this introduced duplicates (e.g., "NHL" vs. "Non-Hodgkin lymphoma"). In order to merge different memos into one corpus, we need to ensure consistency. So, we developed an alias resolution agent that sequentially checks whether a newly parsed indication matches an existing alias and merges competitor lists when appropriate. Domain experts then manually verified and corrected the resulting mappings. For drug-level deduplication, we relied on the LLM judge described in "Alias resolution via an LLM judge" at page 6.

After cleaning and merging, we retain 105 unique indications along with a corresponding competitor drug names lists.

Parser Benchmarking

We split the memos into two non-overlapping batches: 5 memos for prompt refinement and 10 unseen memos for final assessment. On the 5-memo batch we iteratively adjusted the agent configuration and prompt until the parser reproduced every author-annotated drug-asset → indication → competitor triple. We then froze the prompt and ran it once on the 10-memo batch, where it again attained 100 % extraction accuracy; this confirmed the parser and produced the benchmark dataset used in all subsequent experiments.

Data Splitting

We needed a smaller test set to manually run OpenAI Deep Research and Perplexity Labs in order to mimic the behavior of our clients using the corresponding systems through the publicly available UI apps. To prioritize newer indications, we sorted all normalized entries by the latest mention date across their aliases. The 50 most recent were assigned to the test set, and the next 50 to the validation set (see "Validation/Test Split" in Fig. 2). In Fig. 3 we grouped indications into general disease groups and show the distribution over test and valid splits.

Competitor-Validator Dataset: Negative Samples Mining

As mentioned in the earlier sections, the resulting set of indication–competitor drug pairs can be used only to measure the recall. While this list is incomplete, it is sufficient for evaluating coverage. To compute precision, however, we
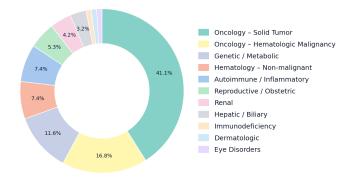

Figure 3: Distribution of indications clustered to general categories.

Definition of a Competitor Drug. Exclude drugs that are completely mechanistically or clinically irrelevant to the given indication, even for learning-from-failure purposes. Drugs that should be 100% excluded include those with no relevant MoA (Mechanism of Action), no trial history, or no theoretical relevance to the disease.

Figure 4: Definition of a Competitor Drug

must also identify false positives among the predicted competitors — drugs that were incorrectly placed in the competitive landscape by the agent.

This requires a scalable evaluation setup that can be integrated into Continuous Integration/Continuous Delivery (CI/CD) pipelines, enabling recall and precision to be computed automatically whenever a new agent is pushed to the code base. We therefore needed a real-time grading mechanism for competitor validity. Specifically, we require a Competitor-Validator Agent that, given an indication and a drug name, could determine whether the drug belongs to the competitive landscape of that indication.

At this stage, we had only positive samples $\mathcal{C}_j^\star$. To iteratively refine the Competitor-Validator prompt, we also required negative samples. For each of the 50 test and 50 validation indications, we used the GPT o3-pro agent to predict competitor drug names. After removing known true competitors, we randomly sampled two remaining candidates per indication. Each resulting pair $(ind_i, drug_i)$ was passed to the original authors of the memos for expert labeling (see "Negative Sample Mining" in Fig. 2.). The task was to determine whether $drug_i$ is a valid competitor for ind_i, following the criterion defined in Fig. 4.

Competitor-Validator Agent

To maximize precision for a predicted competitor list $\widehat{\mathcal{C}}_i$ for a given indication ind_j, we classify each predicted drug $d \in \widehat{\mathcal{C}}_i$ as a true or false positive. This classification is performed by our *Competitor-Validator Agent* (see "Competitor-Validator" in Fig. 2), an LLM-as-a-judge. We use it as a filtering layer to suppress hallucinations and remove false positive drugs from the list of predicted competitors before propagating them further to the client-facing sys-

tem components. This filter maintains both high recall and high precision.

Implementation

The Validation Agent was developed by iteratively refining prompts on the expert-labeled dataset described in Subsection "Competitor-Validator Dataset: Negative Samples Mining" at page 4. The final agent is based on the REACT (Reasoning and Acting) framework with 3 attempts (REACT-3) and utilizes Gemini-2.5 Flash with a web search tool.

The agent's investigation is highly structured. It is prompted to act as a specialist in pharmaceutical market analysis and is instructed to query a range of authoritative sources, such as:

- Clinical trial registries (e.g., ClinicalTrials.gov)
- Regulatory filings and approvals (e.g., from the FDA and EMA)
- Scientific literature and conference abstracts
- Market research reports and investor presentations
- Company press releases on partnerships, licensing, or M&A activity

Crucially, the agent's final judgment is bound by a strict set of rules. It classifies a drug as a competitor only if there is verifiable evidence for the same indication: i) clinical development or approval (including active, completed, failed/discontinued/withdrawn trials and case reports), ii) when no clinical trial history exists, clear mechanistic relevance plus publicly documented IND-enabling and/or pre-clinical evidence. Conversely, it excludes drugs with only theoretical mechanistic relevance without pre-clinical/clinical evidence, entries with wrong/mixed-up identifiers, broad platforms without a specific candidate, and programs for related but distinct indications or different lines of therapy without a clear, direct mechanistic link.

Given an $(indication, drug)$ pair, the agent follows a Thought-Action-Observation loop for *three* iterations to apply these rules and consult these sources. In each cycle, it:

1. **Thinks**: Analyzes previous search results and its current understanding to formulate a hypothesis or identify knowledge gaps.
2. **Acts**: Formulates a targeted web search query based on its thought process.
3. **Observes**: Executes the search, retrieves information, and synthesizes the findings.

This iterative approach enables the agent to build a robust evidence base before making a final judgment. The final output is a JSON object specifying a boolean $\hat{y}_i$ label and a justification. As shown in Table 1, the agent achieves 90.4% precision and 85.7% recall on the test set, resulting in an F1-score of 88.0% and demonstrating its reliability for automated precision grading.

Agents & Models

We evaluated a range of models, grouped into four main categories, to benchmark performance on the competitor discovery task. All models were run with a temperature of

Split	Prec. (%)	Rec. (%)	F1 (%)
Validation	90.7	89.5	90.1
Test	90.4	85.7	88.0

Table 1: Performance of the Competitor-Validator on validation and test splits.

0.0 where supported. We used the same prompts for base models with and without web access, but employed different, specialized prompts for Deep Research agents and the LLM scaffolding frameworks to best leverage their capabilities. All systems were generally tasked to perform thorough research on competitive landscape analysis by covering sources at least mentioned in subsection "Implementation" of section "Competitor Validator" at page 5.

The evaluated model architectures are:

1. **Models without web browsing:** Foundation models relying on their training-time knowledge, including 'o3-pro' and 'gemini-2.5-pro' with web browsing disabled.
2. **Models with web browsing:** Standard API calls to foundation models with native tool-use for web search, including 'gpt-4o', 'gemini-2.5-pro', and 'gemini-2.5-flash'.
3. **Deep Research agents:** Systems optimized for persistent web browsing and integrated reasoning, such as Perplexity and OpenAI Deep Research.
4. **LLM Scaffolding Agents:** To facilitate more complex, multi-step reasoning, we implemented two agentic frameworks on top of LLM calls: REACT, REACT with REFLEXION

REACT: To move beyond single-turn queries, we first implemented a REACT agent (Yao et al. 2023). Our implementation prompts the agent to act as a "diligent and creative research agent" tasked with comprehensively mapping the competitive landscape. The agent iterates through a Thought-Action-Observation loop for a set number of attempts (N). In each loop, it analyzes its previous findings to form a search strategy, formulates a targeted web query, and synthesizes the results, allowing it to dynamically build on its intermediate findings. We denote such configurations with the suffix "$-N$" (e.g., REACT-3 uses 3 REACT iterations/attempts).

REFLEXION: Our experiments with REACT revealed that while increasing the number of iterations improved recall, it often did so at the cost of lower precision. To address this trade-off, we introduced a meta-cognitive layer by implementing a REACT with REFLEXION framework (Shinn et al. 2023). In this two-stage process, a REACT "actor" agent first generates a solution. A second "reflector" agent, prompted as an expert analyst, then critiques the actor's output. The reflector is tasked with diagnosing failures, identifying incorrect competitors, and formulating a high-level plan for improvement. This structured critique is fed back to the actor to guide the next iteration, creating a loop designed to prune false positives and repeated for M rounds. Where the history of actor's answers and reflections to them is propagated to

the next round. We denote such configurations with the suffix "-REFLEXION-M" (e.g., REACT-3-REFLEXION-3 uses 3 REACT iterations/attempts and 3 REFLEXION rounds).

REFLEXION with HISTORY: To further enhance the efficacy of this critique, we experimented with a variant, REACT with REFLEXION-HISTORY. In this setup, we provide the reflector not just with the actor's final output but also its entire reasoning trace — the full history of thoughts, actions, and observations. We hypothesized that this deeper contextual information would allow the reflector to diagnose more subtle flaws in the actor's search strategy and provide a more effective and nuanced critique to guide the subsequent attempt. We denote such configurations with the suffix "-HISTORY" (e.g., REACT-3-REFLEXION-3-HISTORY uses 3 REACT iterations/attempts and 3 REFLEXION rounds and a History).

Web tools: *Observation* step of the REACT accepts the query and performs the query using the web tool specified by the *Act* step. We expose two retrieval tools and let the agent choose among them per step: (i) *Gemini-2.5 Pro* with browsing, and (ii) *Perplexity Sonar*. To ensure that retrieved evidence is genuinely web-sourced and reliable, we pass all candidate snippets through a basic verification gate: checking whether there are groundings returned with the result, whether groundings are not pure text generations with no supports, URL resolution and recovery of the cited span (verbatim or high lexical overlap), cross-source corroboration for high-impact claims (at least two independent sources), deduplication/canonicalization of mirrors and press-wire duplicates.

Step-parallel queries: To increase retrieval *breadth*, we enable step-parallel fan-out: at each REACT step, the agent may generate up to S queries *in parallel*, sending each query to two search tools (Gemini-2.5 Pro, Perplexity Sonar). All returned results are merged in the *Final Answer* step, which consolidates the full history across all steps. We denote such configurations with the suffix "$-S-S_{\max}$"; e.g., REACT-12-S-20 uses 12 REACT iterations and allows up to 20 parallel web searches per step.

Step summary. All "$-S-S_{\max}$" variants include a *step-summary* mechanism. After each REACT step, we extract a compact, structured record of facts and append it to the running state. For competitor discovery, this record captures the canonical drug names identified so far with their associated retrieved insights. These summaries normalize the otherwise heterogeneous, verbose tool outputs, reduce omission/forgetting in long histories. In ablations, agents with step summaries perform better than identical agents without them (see Appendix).

Evaluation

We use LLM-as-a-judge graders for both Competitors Discovery and Attributes Extraction tasks. Below, we describe these graders.

Competitors Discovery Grader

For each indication i we compare the agent's predicted list of competitor drug names $\widehat{C}_i$ with an expert-curated ground-truth list $C_i^\star$. Each ground-truth drug $d \in C_i^\star$ is scored

$$r(d) = \mathbf{1}[d \in \widehat{C}_i],$$

and sample-level recall is $R_i = \frac{1}{|C_i^\star|} \sum_d r(d)$. Mean recall over the benchmark yields the grader output $\mathrm{Recall} = \frac{1}{N} \sum_{i=1}^N R_i$.

Alias resolution via an LLM judge: Drug names in $\widehat{C}_i$ are noisy: a single API may surface as development codes, regional brands, or salt/route strings. No single controlled vocabulary, whether RxNorm, DrugBank, Martindale, WHO Drug Dictionary, or EMA xEVMPD, covers the full alias tail. For example, an FDA normalization of 2024 FAERS opioid reports collapsed 7,892 heterogeneous free-text strings to just 92 RxNorm ingredients after multiple API look-ups and manual edits — evidence that static synonym tables buckle under real-world variability. Rule-based matching undercounts true positives in these conditions.

To avoid such alias-resolution complexity, we delegate drug alias resolution to a Gemini-2.5 Pro LLM judge with temperature set to 0. For example, for a single drug name `Utrogestan` from the ground-truth list and a predicted set `[Atosiban, Diclofenac, ..., Nitroglycerin, Progesterone]`, the LLM judge returns the structured verdict in Listing 1.

Attributes Extraction Graders

We evaluate both indication-independent and indication-dependent attributes with the same LLM-as-a-judge protocol as in Competitors Discovery: all graders call *Gemini-2.5 Pro* at $temperature = 0$; where attribute grading requires external evidence, the judge is run with web browsing.

Drug name aliases grader validates each predicted alias via web-grounded checks. The score it outputs is a precision over the validated set:

$$\mathrm{Prec} = \frac{|A_{\text{correctly predicted}}|}{|A_{\text{all predicted}}|}.$$

Modality grader checks semantic match between the predicted modality and the reference. The score is the binary accuracy per sample.

Lead indication grader tests whether the predicted lead indication semantically matches ground truth (GT). The score is the binary accuracy.

Listing 1: Structured verdict returned by the Gemini-2.5 Pro LLM judge for drug aliases resolution.

```
1  {
2    "Utrogestan": {
3      "present": true,
4      "matched_aliases": ["Progesterone"],
5      "reason": "The drug 'Utrogestan' is
         a brand name for the generic drug
         Progesterone. The predicted list
         includes 'Progesterone', which
         is considered a match."
6    }
7  }
```

Administration route grader checks whether the GT route is in the predicted set applying synonym handling. The score is the binary accuracy.

Other indications. For each drug, compare the ground-truth set of non-lead indications to the system's predicted set. The per-example recall is

$$Rec = \frac{1}{|\mathcal{I}|} \sum_{u \in \mathcal{I}} \mathbf{1}\left[u \in \widehat{\mathcal{I}}\right],$$

where $\mathcal{I}$ is the ground-truth set of "other" indications, $\widehat{\mathcal{I}}$ is the predicted set, u indexes indications, and $\mathbf{1}[\cdot]$ is the indicator function. (Reported score is the mean of R over the benchmark.)

Mechanism of action (MoA) grader decomposes the ground truth MoA into atomic statements and checks whether each statement is expressed (paraphrastically) in the prediction. The score is the statement coverage

$$C = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}\{\text{statement}_j \text{ present}\}.$$

Target grader normalizes names/aliases and checks if any predicted target matches any GT target. The score is the binary accuracy.

Development stage & regulatory status grader compares the prediction to the most advanced historical trial state mentioned in the memo. Predictions must be $\geq$ of that state in the clinical progression lattice and status order. The score is the binary accuracy.

Therapeutic area (TA) grader performs the web-grounded verification that the predicted TA matches GT for the specific drug/indication. The score is the binary accuracy.

Company information Company information is the composite field with *three* sub-attributes: official website URL, company description, and stock ticker. Web-grounded, cascaded check: if the website is wrong, the whole attribute scores 0; otherwise the score is the mean of website correctness, ticker correctness, and description validity (the latter is the fraction of description statements validated online).

The score is computed the following way:

$$s = \begin{cases} 0, & \text{if website is incorrect,} \\ \dfrac{s_{\text{site}} + s_{\text{ticker}} + s_{\text{desc}}}{3}, & \text{otherwise.} \end{cases}$$

Benchmarking

Competitors Discovery

In this subsection, we show that specialized agents, tuned for a specific task – in this case, competitor discovery – outperform general-purpose AI systems. We then illustrate that although LLMs without web access can achieve high recall, removing easy samples from the test set causes recall to decline. In contrast, multi-hop LLM scaffolding with web-tool usage becomes increasingly important, showing significantly less degradation.

Comparison Across Models In Table 2, we report only our three best-performing agents. For the full list of experiments with comprehensive analysis, refer to the subsectoin "Ablation Study of Competitors Discovery" in Appendix at page 9. Also, here we report production scenario, where before returning competitors's list and hence computing recall, we suppress all false positives predicted by Competitor-Validator.

The table shows that our variant of REACT-12-S-20 outperforms OpenAI Deep Research and Perplexity Labs, as well as o3-pro with no web, gpt-5, gpt-4o, gemini-2.5-pro. OpenAI Deep Research and Perplexity Labs were evaluated through their publicly available UI applications. In contrast, o3-pro, gpt-5, gpt-4o, gemini-2.5-pro were evaluated via API access; note that o3-pro does not use web search, as the API version did not support this at the time of writing.

REACT-12-S-20 is run with temperature 0, to ensure determinism, however when we switch to gemini-2.5-pro's default temperature 1.0 and run 3 REACT-12-S-20 agents in parallel and merge their results (REACT-12-S-20-ENSEMBLE-3), it gives the best recall of 0.83.

System	Recall
ReAct-12-S-20-Ensemble-3 (gemini-2.5-flash)	**0.83**
ReAct-12-S-20 (gemini-2.5-pro)	0.78
ReAct-3-Reflexion-3-History (gemini-2.5-pro)	0.77
ReAct-3 (claude-sonnet-4)	0.68
o3-pro (no web)	0.67
OpenAI Deep Research	0.65
gpt-5	0.63
Perplexity Labs	0.60
gemini-2.5-pro	0.59
gpt-4o	0.56

Table 2: **Competitors Discovery performance comparison** across systems on test split of the Competitors Discovery Dataset. Recall is given after predicted false positives were suppressed by Competitor-Validator.

Performance on Hard Samples To evaluate model robustness, we analyzed performance across increasingly difficult indications. We define difficulty relative to the performance of the non-web o3-pro baseline: samples on which this model performs poorly are considered more difficult. As shown in Figure 1, the performance of non-web models collapses on these harder samples, confirming that grounding in external knowledge is essential.

The analysis further shows that the performance gap between simple web-enabled models and the iterative REACT agents widens as task difficulty increases. This suggests that for difficult indications where information is fragmented, the ability to conduct multi-hop reasoning by synthesizing evidence across multiple web search attempts is critical. In the most difficult regimes (e.g., performance of o3-pro is $\leq 40\%$), a higher number of search iterations (REACT-5) provides a clear advantage over fewer iterations, underscoring the value of persistence. Throughout all difficulty levels, 'gemini-2.5-flash' remains competitive with or outperforms 'gemini-2.5-pro', indicating that the model size is less sig-

nificant when the model is allowed to perform a few web searches.

Attributes Extraction

Table 3 reports per-attribute scores using the graders (see "*Attributes Extraction Graders*" subsection at page 6). A lightly tuned REACT-12 with expert-curated prompts performs on par with OpenAI Deep Research for drug-attribute extraction, matching it on MoA and Target while edging it on Development Stage/Status, Therapeutic Area, and Company. MoA remains the most error-prone field for all systems. Performance is strong overall – perfect on Modality and high on Administration Route – while Lead Indication and Aliases are lower.

Attribute	OpenAI Deep Research	ReAct-12
Overall	0.76	**0.82**
Aliases	0.78	0.79
Modality	0.96	1.00
Lead indication	0.80	0.76
Administration route	0.90	0.91
Other indications	0.14	0.43
MoA	0.61	0.61
Targets	0.84	0.84
Status & Stage	0.84	0.92
Therapeutic area	0.92	1.00
Company info	0.77	0.89

Table 3: **Attributes extraction performance** (higher is better) with models as columns. See subsection "Attributes Extraction Graders" on page 6 for metric definitions and scoring rules.

Production

Operational impact. In a private biotech VC case study, instrumenting analyst workflows for competitive scans yielded a substantial throughput gain: analyst turnaround time *per drug asset* decreased from ~ 2.5 days to ~ 3 hours ($\approx 20\times$ faster) while maintaining analyst-verified precision. On the Competitors Discovery Benchmark (Table 2), the recall – coverage of drug asset names $\mathcal{C}_i^\star$ identified by experts as in the competitive environment for the given indication – is not 100%; however, more importantly, the system surfaced additional relevant drug assets outside that ground truth set $\mathcal{C}_i^\star$. After review, VC analysts validated these previously unseen assets as correct and decision-useful, indicating improved discovery power without sacrificing precision.

Deployment We deploy our REACT agent behind a lightweight Gradio front end to support analyst-in-the-loop review. The back end is a graph-orchestrated agent service, where nodes define the agent's logic and edges determine execution flow.

Operationally, we enforce per-tool timeouts and budgets, per-tenant rate limits, version-controlled prompts and datasets, and structured logging/metrics. We also integrate Competitor Discovery and Attribute Extraction evaluations into our CI/CD process.

Conclusion

In this paper we present a deployed, domain-grounded system for competitor discovery in drug asset due diligence, pairing a high-recall REACT-style web agent with an LLM-as-judge "Competitor-Validator". In a benchmark derived from real biopharma VC diligence memos, the agent outperforms general-purpose systems in recall, while postprocessing based on 'Competitor-Validator' suppresses false positives without sacrificing precision.

Robustness analyses (see subsection "Performance on hard samples" at page 7; Fig. 1) show that non-web models and single-pass prompting agents degrade on harder indications, whereas scaffolded agents with multi-hop web search maintain performance. Because the difficulty metric proxies long-tail, under-linked competitors—the cases that drive diligence risk—this implies that the agent's gains are largest where they matter most. Although certain single-pass agents achieve higher recall in our ablations (see Table 4 in "Appendix"), this difficulty-stratified analysis shows these wins concentrate on easy cases; their recall drops sharply on hard samples, whereas scaffolded, web-search agents maintain performance.

In production, the system substantially reduces analyst turnaround and surfaces decision-useful competitors beyond historical ground truth.

References

Benaïche, A.; Billaut-Laden, I.; Randriamihaja, H.; and Bertocchio, J.-P. 2025. Assessment of the Efficiency of a ChatGPT-Based Tool, MyGenAssist, in an Industry Pharmacovigilance Department for Case Documentation: Cross-Over Study. *Journal of Medical Internet Research*, 27: e65651.

Biran, E.; Gottesman, D.; Yang, S.; Geva, M.; and Globerson, A. 2024. Hopping Too Late: Exploring the Limitations of Large Language Models on Multi-Hop Queries. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 14113–14130. Miami, Florida, USA: Association for Computational Linguistics.

Chen, Q.; Hu, Y.; Peng, X.; Xie, Q.; Jin, Q.; et al. 2025. Benchmarking large language models for biomedical natural language processing applications and recommendations. *Nature Communications*, 16: 3280.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36: 28091–28114.

European Commission. 2025. Joint Clinical Assessments. Scope from 12 Jan 2025 includes new cancer medicines and ATMPs.

FDA. 2023. Demonstrating Substantial Evidence of Effectiveness With One Adequate and Well-Controlled Clinical Investigation and Confirmatory Evidence (Draft).

Gema, A. P.; Hägele, A.; Chen, R.; Arditi, A.; Goldman-Wetzler, J.; Fraser-Taliente, K.; Sleight, H.; Petrini, L.; Michael, J.; Alex, B.; Minervini, P.; Chen, Y.; Benton, J.; and Perez, E. 2025. Inverse Scaling in Test-Time Compute. arXiv:2507.14417.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. arXiv:2401.13919.

ICH. 2000. ICH E10: Choice of Control Group and Related Issues in Clinical Trials.

Lù, X. H.; Kasner, Z.; and Reddy, S. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

McDuff, D.; et al. 2025. Towards accurate differential diagnosis with large language models. *Nature*.

NICE. 2022. *NICE health technology evaluations: the manual (PMG36).* Comparator(s) must reflect routine NHS practice and be justified.

OpenAI. 2024. Accelerating the development of life-saving treatments: Moderna partners with OpenAI and deploys ChatGPT Enterprise. OpenAI Case Study (accessed Aug 2025).

PwC Germany. 2024. Case study: Takeda — Building TAK GPT on Microsoft Azure OpenAI Service. PwC case study.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K. R.; and Yao, S. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Starace, G.; Jaffe, O.; Sherburn, D.; Aung, J.; Chan, J. S.; Maksin, L.; Dias, R.; Mays, E.; Kinsella, B.; Thompson, W.; et al. 2025. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848*.

Wei, J.; Sun, Z.; Papay, S.; McKinney, S.; Han, J.; Fulford, I.; Chung, H.; Passos, A. T.; Fedus, W.; and Glaese, A. 2025. BrowseComp: A Simple yet Challenging Benchmark for Browsing Agents. *arXiv preprint arXiv:2504.12516*.

Wu, Y.; Huang, Y.; Du, Q.; Lai, L.; He, Z.; Hu, J.; and Tao, X. 2025. Should I Believe in What Medical AI Says? A Chinese Benchmark for Medication Based on Knowledge and Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 1155–1164. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-252-7.

xAI. 2025. Grok 4 Heavy: Parallel Test-Time Compute for Multi-Agent Tool Use. https://x.ai/news/grok-4. Accessed 29 July 2025.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Appendix

Ablation Study of Competitors Discovery

Table 4 is computed in an unfiltered evaluation regime: the Competitor-Validator is not used to suppress model outputs; instead, its signal contributes only to the precision calculation. This differs from Table 2, which reflects our deployment setting where Competitor-Validator acts as a conservative acceptance policy (low-confidence predictions are dropped). As expected when removing a conservative filter, recall increases in Table 4, while precision reflects Competitor-Validator's judgment rather than pre-screening. Because the evaluation regimes (and underlying systems) differ, the two tables are not directly comparable; together they surface the precision–recall trade-off between a deployment-ready setting (filtered) and an analysis setting that measures upper-bound extraction capacity (unfiltered).

The REACT with REFLEXION configurations consistently deliver the highest F1 scores, with the top variant achieving 0.84. This demonstrates that a meta-cognitive loop for self-correction is a decisive factor in achieving robust performance. Our scaffolding framework also proves to be versatile: the REACT-3 + REFLEXION-3-HISTORY agent attains the second highest recall (0.841), positioning it as a powerful "explorer" for comprehensive discovery. Other configurations, such as REACT-1-REFLEXION-6, REACT-1-REFLEXION-3-S-10 can be tuned for high precision (0.87/0.95), rivaling specialized systems. While dedicated search agents like OpenAI Deep Research can achieve the benchmark's highest precision (0.935), this specialization comes at a significant cost to recall, resulting in a lower overall F1 score than our top REFLEXION agents.

Beyond headline scores, Table 4 shows three consistent patterns. First, adding step-parallel search and modest ensembling improves coverage without a large precision penalty: REACT-12-S-20-ENSEMBLE-3 attains R=0.845, P=0.900, F1=0.871, indicating that breadth of retrieval—not just deeper reasoning—drives recall. Second, REFLEXION stabilizes the precision–recall trade-off at small N: REACT-1-REFLEXION-3-S-10 reaches the highest precision (0.95) at the cost of recall (0.740; F1=0.842), whereas the history variant REACT-3-REFLEXION-3-HISTORY maximizes recall among REFLEXION settings (0.841) by leveraging the actor/reflector trace. Third, simply increasing iteration budgets without adding a step summary (all '$-S-S_{max}$' variants use step summary) shows diminishing or negative returns, due to context overload: REACT-30-REFLEXION-3 raises recall (0.815) but depresses precision (0.695), lowering F1 to 0.750. General-purpose systems remain skewed—e.g., OpenAI Deep Research (P=0.933, R=0.671; F1=0.780) and gpt-5 (P=0.835, R=0.636; F1=0.722)—underscoring the value of targeted scaffolds

Model	Web	Precision	Recall	F1
ReAct-12-Reflexion-3-S-10 (gemini-2.5-pro)	✓	0.933	0.840	**0.884**
ReAct-12-S-20-Ensemble-3 (gemini-2.5-flash)	✓	0.900	**0.845**	0.872
ReAct-12 (gemini-2.5-pro)	✓	0.865	0.844	0.854
ReAct-1-Reflexion-3-S-10 (gemini-2.5-pro)	✓	**0.950**	0.740	0.832
ReAct-1-Reflexion-3 (gemini-2.5-pro)	✓	0.864	0.798	0.830
ReAct-1-Reflexion-6 (gemini-2.5-pro)	✓	0.870	0.768	0.816
ReAct-3-Reflexion-3 (gemini-2.5-pro)	✓	0.800	0.815	0.807
ReAct-3-Reflexion-3-History (gemini-2.5-pro)	✓	0.773	0.841	0.806
ReAct-1 (gemini-2.5-pro)	✓	0.834	0.742	0.785
OpenAI Deep Research	✓	0.933	0.671	0.781
ReAct-5 (gemini-2.5-pro)	✓	0.757	0.789	0.773
gemini-2.5-pro	✓	0.852	0.699	0.768
o3-pro (2 candidates)	✗	0.721	0.796	0.757
ReAct-30-Reflexion-3 (gemini-2.5-pro)	✓	0.695	0.815	0.750
ReAct-3 (gemini-2.5-pro)	✓	0.727	0.773	0.749
gemini-2.5-pro	✗	0.814	0.691	0.747
4o	✓	0.910	0.608	0.729
ReAct-3 (claude-sonnet-4)	✓	0.730	0.727	0.728
Perplexity Labs	✓	0.865	0.629	0.728
gpt-5	✓	0.835	0.636	0.722
ReAct-1 (gemini-2.5-flash)	✓	0.629	0.838	0.719

Table 4: **Comparison of LLM agents performance** with and without web search. Agents are evaluated by Precision, Recall, and F1 on a test split of a Competitors Discovery Dataset. Results are sorted by F1. No predictions post-filtering with Competitor-Validator.

with verification. Practically, these results justify pairing a high-recall explorer (e.g., the history variant at small N or REACT-12-S-20) with the Competitor-Validator to enforce precision, and reserving high-precision REFLEXION settings for verification passes.