SID-2 Technical Report: Scaling Search Swarms

SID Research

research@sid.ai

Abstract

We introduce our second-generation search models, SID-2 and SID-2 Swarm. We trained our first-generation model, SID-1, with reinforcement learning to take a question, iteratively search, read results, and finally report a ranked list of relevant documents. This made search adaptive but came with two major limitations. SID-1 and SID-2 act as single search agents. Although they can issue queries in parallel, incremental turns add latency, and retrieved context accumulates in a single context window. SID-2 Swarm is trained as a team of agents that divide and conquer. They coordinate, fork into separate contexts, and share findings. On a benchmark of diverse, difficult search questions, SID-2 at medium effort reaches 0.58 NDCG in 3.1 s of model inference, matching what Claude Opus 5.5 at medium effort achieves in 25 s. With six agents, SID-2 Swarm reaches 0.70 NDCG in 18 s, matching what GPT-6 Astra at maximum effort achieves in 135 s. We find that coordinated swarms improve retrieval quality per FLOP more than fusing results from independent runs or increasing the model's number of parameters. We believe this is due to search benefiting more from exploring a wider range of documents than it does from reasoning more deeply about each one.

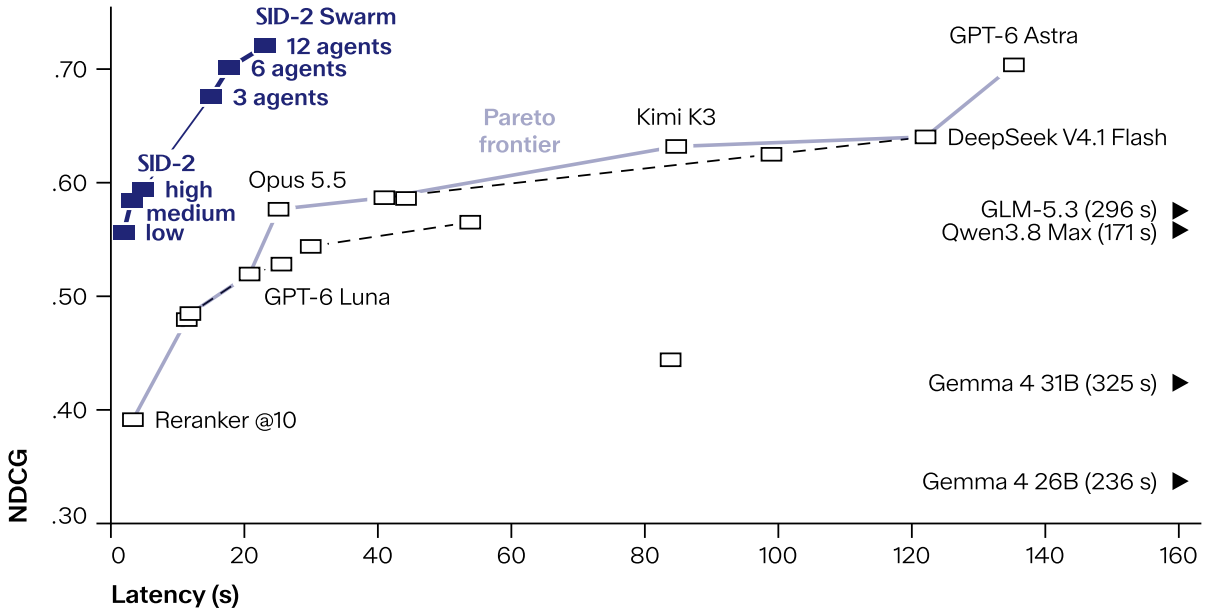

Figure 1 | Retrieval quality (NDCG) and model latency (calculated).

Contents

1 Introduction **3**

2 Capabilities **4**

3 SID-2 **5**

3.1 Metric Steering 5

3.2 Reasoning effort 6

3.3 Latency model 8

3.4 Proposer-Solver Self Play 10

3.5 Harness 11

3.6 Data 12

4 SID-2 Swarm **13**

4.1 Search parallelizes well 13

4.2 Multi-agent harness design 14

4.3 Simulated-latency request scheduling 15

4.4 Stabilizing entropy for exploration 15

4.5 Emergent collaborative behaviors 16

4.6 Future work 18

5 Infrastructure **18**

5.1 Training optimizations 18

5.2 Train-inference mismatch 18

5.2.1 Responses API for eliminating tokenization mismatches 18

5.2.2 Importance sampling and infrastructural mitigations for numerical mismatch 19

6 Results **20**

7 Conclusion **21**

8 Citation **21**

A Multi-agent coordination transcripts **26**

A.1 Twelve agents on a legal query 26

B Performance by Corpus **30**

1. Introduction

Artificial Intelligence is producing more words per day than human beings write and speak, growing 9–15× year-over-year (Kharazian, 2026). We estimate that, as of November 2026, LLMs are generating as many output tokens as exist on the entire public internet every 15 days. As token costs fall (Emberson and Roodman, 2026) and agent swarms increasingly decouple token generation from human prompting (OpenAI, 2026a,c), we expect this growth to accelerate further. Yet, as with the internet and academic literature, the value of this growing body of information largely hinges on our ability to retrieve it effectively.

As corpora grow, finding relevant evidence becomes harder. Fixed search pipelines built around BM25, embeddings, and rerankers allocate a fixed amount of compute per query, limiting their ability to adapt to query difficulty and corpus size. Predictably, we observe degradation of search quality as a function of corpus size. To improve coverage across corpora of varying sizes and complexities, engineers often add query rewrites, routing rules, and specialized processing. These piecemeal fixes create burdensome complexity and, in some cases, they can undermine each other.

SID-1 (SID Research, 2025) introduced a new search paradigm that replaced hand-written rules with a more flexible, agentic approach learned through reinforcement learning. It makes an initial search, reads the results, and adapts its search strategy as needed. By dynamically varying the number of turns, SID-1 scales inference-time compute alongside query difficulty, improving retrieval across legal, finance, science, and general knowledge domains. Most of the gains come from performance on complex, multi-hop queries where fixed pipelines tend to return only part of the answer. On this evaluation, SID-1's agentic approach achieves higher recall overall than fixed retrieval pipelines (Figure 2).

While SID-1 improved recall over fixed pipelines, its scalability remained constrained by latency and its effective context window. Additional turns add latency and accumulate observations in the same context window, effectively capping the volume of evidence that the model can examine. We therefore hypothesized that search would benefit more from division of agentic labor than other domains, such as math and coding. While many other tasks require sequential reasoning over intermediate results, a corpus can be easily divided across agents and searched in parallel. SID-1 demonstrated gains from combining four independent searches and identified coordination as a priority for future work.

SID-2 uses agent swarms to coordinate parallelized searches. Through reinforcement learning, the agents learn to explore in separate contexts and share discoveries that inform subsequent searches. This expands the breadth of evidence examined beyond one agent's effective context window and allows searches to fan out concurrently. In our evaluations, coordinated groups of small models outperform larger, more capable individual models, including in reasoning-heavy domains like legal and finance. The Bitter Lesson (Sutton, 2019) does not necessarily call for larger models, but rather methods that scale effectively with compute.

Of course, broader exploration consumes additional compute and may return more evidence than a given use case requires. We therefore train SID-2 for configurable reasoning effort and *Metric Steering*, allowing users to control the tradeoff between quality and latency and specify their retrieval objective.

Through compounding returns to data, post-training methods and model architecture we drastically increase our model's efficiency. At its lowest effort setting, SID-2 outperforms four parallel SID-1 runs using 13× fewer FLOPs per task.

Part of the success of coding agents stems from their ability to complete long-running tasks autonomously. They can inspect a repository, read relevant files, and continue working without waiting for a human to assemble the necessary context. We believe that knowledge work agents in other domains have yet to achieve the same degree of self-sufficiency in part because they face larger and less structured search spaces. SID-2 seeks to address this.

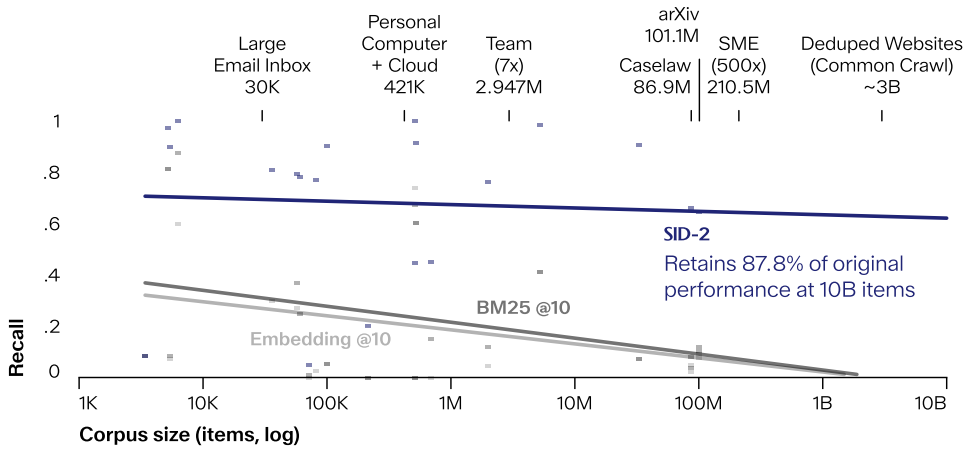

Figure 2 | Recall against corpus size for SID-2 Swarm (12 agents), SID-1, embedding search @10 and BM25 @10. Points are dataset means, and solid lines are fits against logarithmic item count with equal weight per dataset. Extrapolating each fit to 10B items, SID-2 Swarm retains 94.0% of the recall it has on the smallest corpora (<10k items) and SID-1 retains 15.3%; the embedding and BM25 fits reach zero before 10B items. All methods use the same 506 questions; agent reports are unrestricted, while retrieval baselines return their top 10 candidates.

2. Capabilities

In service of raw retrieval performance and new use cases we discovered in deployment of SID-1, we train a number of new *capabilities* into SID-2: new ways in which the model can be adapted and scaled in deployment.

Agent swarms (§4). For best-in-class retrieval performance, we train SID-2 Swarm, whereby multiple agents can message each other and fork their context into more agents. We find this to be a very promising direction for the problem of search specifically.

Very low latency settings (§3.2). We train SID-2 to have an extremely low reasoning effort mode, while achieving comparable performance to models with 1–2 orders of magnitude more parameters on high reasoning settings.

Long context. SID-2 supports a 262k-token context window, and we train on long-context retrieval tasks in our high reasoning effort settings.

Metric Steering (§3.1). We allow users to steer the breadth vs depth tradeoff in the documents returned by SID-2 and SID-2 Swarm by natively training the model to be steerable to different points on the precision-recall curve.

Complex, multi-corpus environments (§3.6). We dedicate significant effort to creating training environments with complex, interlinked corpora with different search tool configurations and filters, based on our insights from deploying SID-1.

3. SID-2

3.1. Metric Steering

Part of what makes search challenging is that different applications call for different degrees of depth and breadth. For example, a paralegal reviewing discovery files would prioritize recall (comprehensiveness) over precision, because, in this case, missing a critical piece of evidence is more costly than skimming an extraneous document. By contrast, an e-commerce shopper searching for a '40W incandescent lightbulb' is unlikely to scroll past the first handful of results and therefore values precision more highly. These two users have opposing priorities but are likely given similar tools and see similar results today, depending on corpus sizes. This is because classical search tools like BM25, embeddings, and even SID-1, provide little latitude to hone the recall/precision tradeoff. Rerankers, query rewriters and other techniques can help mitigate this problem, but they do not address the underlying cause.

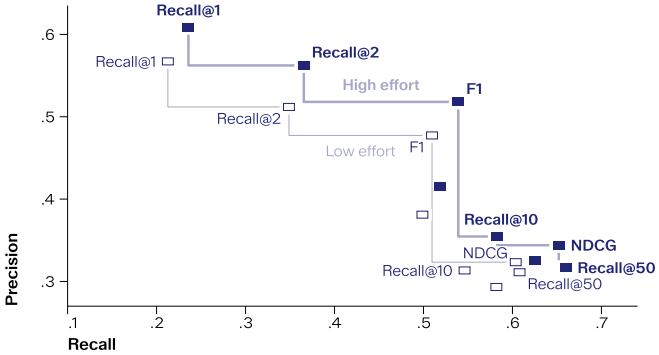

Figure 3 | Precision against recall of SID-2 on 506-question validation dataset for each metric steering prompt (Recall@k, F1 and NDCG), at low and high reasoning effort. Each line is the Pareto frontier of one effort level, joining the prompts that no other prompt beats on both recall and precision; only these are named.

We therefore introduce *Metric Steering* during post-training, that allows users to specify their primary retrieval objective and steer the model to optimize for it directly within the search loop. SID-2 learns to adapt its search strategy, context management, and reporting behavior to the desired reward metric at inference time. It can natively optimize for the following metrics:

- **NDCG.** Comprehensive search that does not penalize reporting additional background documents (examples: legal, financial or scientific research, compliance).
- **F1.** High-precision, penalizes reporting non-critical documents (examples: job search, customer support, e-commerce).
- **Recall@k.** Find exactly $k \in \{1, 2, 5, 10, 20, 50\}$ items (examples: workflows, agents with context management, curated chat bars that recommend a preset number of results).

Metric Steering is one of agentic search's primary advantages over traditional methods. Previously, engineers endured a series of ad hoc retrieval pipeline optimizations to reach their desired point on the recall/precision tradeoff curve. With SID-2, this can be done by modifying an API parameter. In all settings, SID-2 is trained to rank the most relevant documents first.

Metric steering is not just prompting. Shown the same objective prompts, the base model barely changes: its NDCG, F1 and Recall@10 move by at most 0.05, while SID-2 at high reasoning effort scores 0.14 to 0.22 higher on each (Figure 4).

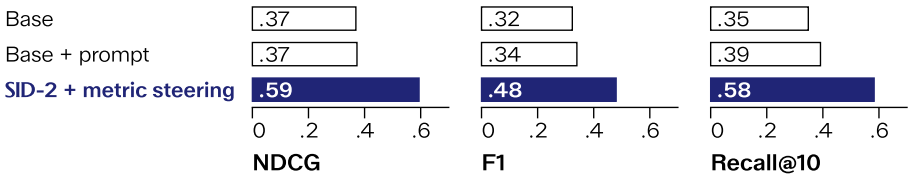

Figure 4 | NDCG, F1 and Recall@10 of the base model without an objective prompt, the base model given the matching objective prompt, and SID-2 with metric steering (given the same prompt) at high reasoning effort. The base model gets no reasoning-effort instruction.

3.2. Reasoning effort

Like the Federal Reserve, SID-2 has a dual mandate with competing objectives: search quality and speed. The behaviors that improve quality (more searches, more reads, more reasoning, more turns) also tend to increase latency. One of our goals with SID-2 is to give users greater control over where they land on the quality/latency frontier. For some retrieval tasks, such as e-commerce search or doc retrieval for live chat, a slow answer is tantamount to a wrong answer. In other domains, such as legal or scientific research, users may gladly invest additional seconds in exchange for higher-quality results.

Prior approaches. Configurable reasoning effort is an obvious way to offer this flexibility and has become standard across many recent models. Implementations take several forms, including binary switches (reasoning on or off), token budgets, and discrete effort levels. DeepSeek trains separate effort specialists, each with its own context window and length penalty (DeepSeek-AI, 2026). Moonshot AI uses per-task token budgets during RL for Kimi K3 (Kimi Team, 2026b). NVIDIA introduces a medium-effort mode during SFT and further optimizes it in post-training (NVIDIA, 2026). Cognition offers a notably explicit approach: SWE-2 subtracts a linear cost penalty from the reward, multiplied by a per-effort coefficient matched to the local slope of the base model's Pareto frontier (Cognition, 2026). Despite the growing predominance of configurable reasoning effort, the literature offers surprisingly little guidance on how to train this behavior into a model. Our goal with SID-2 is to support three effort levels: low, medium, and high. We describe our approach in detail to shed light on how it can be done.

Two-stage training. Initially, we applied latency pressure from the start of training, but this was overly punitive for a model that had not yet learned how to search effectively. It discouraged exploration of valuable (but time-consuming) search strategies that were necessary to achieve maximum quality. A process must be learned before it can be accelerated, so we train SID-2 in two stages. In the first stage, we ignore latency and reward retrieval quality exclusively, allowing the model to experiment wildly, free of latency constraints. In the second stage, once we observe robust search quality, we add a latency penalty, estimated by the *Latency Model* (§3.3). This pushes the model to shed latency-inducing behaviors that yield little or no quality improvement.

Multiplicative latency penalty. Prior work penalizes cost additively, subtracting a length penalty (DeepSeek-AI, 2026) or a linear cost term (Cognition, 2026) from the reward. We instead apply the episode latency L multiplicatively, discounting only the positive part of the reward, r_+:

$$r = \max\left(0,\ 1 - \frac{L}{L_{\max}}\right) r_+ + r_-, \tag{1}$$

where $r_+ \in [0, 1]$ is the retrieval score, $r_- \leq 0$ the penalties, and $L_{\max}$ the latency budget of the prompt's effort level.

With an additive penalty, an episode that retrieves nothing relevant can earn more reward than its peers by finishing sooner, which teaches the model to give up early on hard queries. Under our multiplicative penalty, such an episode receives no latency signal at all: speed only pays off once the model has found something relevant. Finally, the penalties r_- are not discounted, so a fast episode cannot shrink the cost of a malformed or invalid response.

Effort prompts alone. Our naive first approach was to simply state the effort in the system prompt, for example, "Reasoning effort: LOW. Make few tool calls and turns, and minimize your reasoning." While this did modulate performance to a certain extent, it did not do so as much as we had hoped.

Mixed-effort RL. To accentuate differences between effort levels, we incorporate effort prompts into RL training and give each level a different latency budget $L_{\max}$ in the reward, computed by the *Latency Model* (§3.3). This bakes variable effort prompts into the model, but updates for different effort levels undermine each other, so behavior again converges across all settings. Blending reasoning efforts in the same training mix leads to interference between the modes: introducing some low-effort samples draws down the latency of the model when responding to high-effort prompts, even if the reward for these high-effort samples has not changed.

Effort experts. To mitigate this interference, we train a separate expert for each of the three effort levels. Similarly, each expert receives its own latency budget, but, in this case, each settles at a point where the marginal gain in retrieval quality equals the marginal latency cost (Figure 5). We then combine the three experts into a single model. Merging them in weight space either through simple averaging (Wortsman et al., 2022) or through task arithmetic, where a scaling coefficient controls how strongly each expert's parameter delta contributes (Ilharco et al., 2023), scores well on our metrics. However, the merged policies are less stable and show more behavioral outliers, such as repeated sequences and unnecessarily long reasoning.

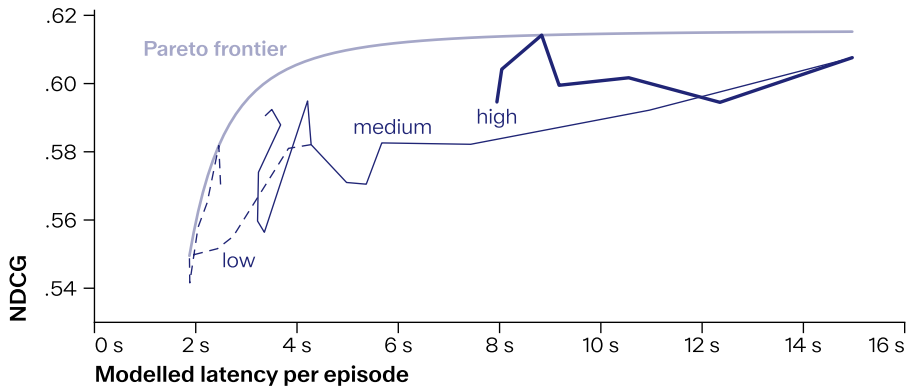

Figure 5 | Mean NDCG on the 506-query evaluation set against mean modeled latency L per episode, over training for the high, medium and low effort runs. Each line joins one effort level's evaluation checkpoints (every 8 RL steps) in training order, starting from the checkpoint it branched off; each effort level is named at its run's last checkpoint. Throughout each run, the latency budget $L_{\max}$ was progressively lowered.

Multi-teacher on-policy distillation. To obtain a more stable policy, we leverage multi-teacher on-policy distillation (MOPD), a technique that has been adopted by several frontier models (Ma et al., 2026a) to combine distinct capabilities in a single set of weights. In MOPD, the student generates episodes, and the corresponding teacher runs a forward pass over the student's token IDs to compute its log probabilities. The reverse KL between the student and teacher distributions provides a dense, per-token training signal.

Figure 6 shows the resulting model at each effort level, one point per question. Within a single effort level, inference time still varies by more than an order of magnitude across questions. The effort level sets how much SID-2 searches overall, but the model still decides per question how long to search, spending more time where the evidence is harder to find instead of using a fixed budget.

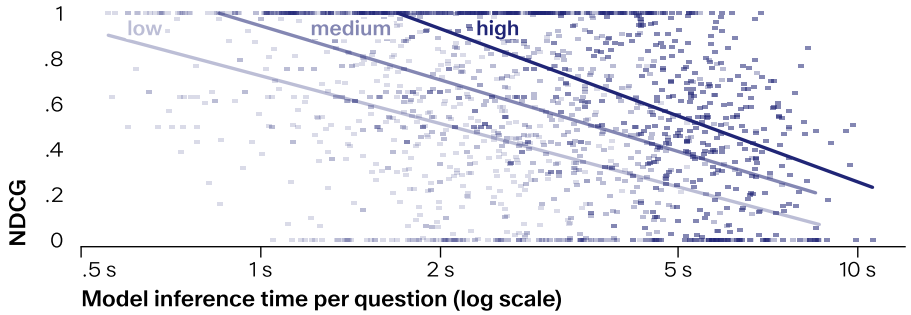

Figure 6 | NDCG against estimated inference time for each of 506 validation questions, for SID-2 at low, medium and high reasoning effort (each point is the mean of 3 samples). Inference time is the latency model's estimate for the question's own token profile on the production stack; retrieval backend time is excluded. Lines are least-squares fits of NDCG on log time for each effort level.

3.3. Latency model

Token counts, which most prior work either budgets or penalizes, are a poor proxy for agentic search latency. All retrieval calls in a turn share a single backend round, prefill tokens are far cheaper than decoded ones, and later turns decode more slowly as the KV cache grows.

Similarly, wall-clock time during post-training is another flawed latency metric, because it is confounded by how an episode is batched and scheduled on the training cluster and is not necessarily indicative of how it would run in production. Two identical episodes could therefore receive different rewards by virtue of scheduling and concurrency.

Instead, we estimate latency analytically, by replaying each episode as a sequence of forward passes that a production server would run. We price each pass on a target serving GPU and add estimated retrieval backend latency. The result is a deterministic latency estimate for each episode, so the reward only changes if the model's behavior changes.

Forward passes. Each model turn becomes one prefill pass over its new input tokens, i.e., the prompt on the first turn and the tool observations on subsequent turns. This is followed by decode passes for output tokens. In serving, we use DFlash (Chen et al., 2026) for speculative decoding (training rollouts use an EAGLE draft head instead, §5.1), so each decode pass emits a tokens on average, and a turn with L_{out} output tokens costs $\lceil L_{\text{out}}/a \rceil$ decode passes.

Pricing a forward pass. In every forward pass, the GPU reads the weights and the KV cache from HBM and performs the matrix calculations. The two operations overlap, so a pass p takes as long as the slower of the two:

$$t(p) = \max\big(t_{\text{mem}}(p),\ t_{\text{comp}}(p)\big), \tag{2}$$

$$t_{\text{mem}}(p) = \frac{W + K(p) + S + (c-1)\,\bar{M}}{\text{BW}}, \tag{3}$$

$$t_{\text{comp}}(p) = \frac{2 \cdot A \cdot T(p) + (c-1)\,\bar{C}}{\text{FLOP/s}}, \tag{4}$$

$$\bar{M} = \frac{1}{P}\sum_{p}\big(K(p) + S\big), \tag{5}$$

$$\bar{C} = \frac{1}{P}\sum_{p} 2 \cdot A \cdot T(p), \tag{6}$$

where

- W is the size of all model weights in bytes.
- $K(p) = k(p) \cdot b_{\text{kv}}$ is the KV cache memory read in pass p, where $k(p)$ is the number of tokens in the KV cache before the pass and b_{kv} the KV cache bytes per token.
- $T(p)$ is the number of tokens processed in pass p. For a prefill pass, this is the number of new input tokens; for a decode pass, it is the number of tokens speculated, of which on average a are generated.
- A is the number of active parameters, used as the FLOPs per token.
- BW is the observed memory bandwidth and FLOP/s the observed floating point operations measured on a serving GPU.
- P is the total number of forward passes in the episode. Each turn i contributes one prefill pass and $\lceil L_{\text{out},i}/a \rceil$ decode passes, so $P = \sum_i \big(1 + \lceil L_{\text{out},i}/a \rceil\big)$.
- c is the prompt concurrency. A production server never runs a request alone, so we make the simplifying assumption that each request shares its batch with $c - 1$ statistically identical copies of itself.
- $\bar{M}$ and $\bar{C}$ are the mean memory traffic and compute per pass that each of the copies adds.
- S is the size of the recurrent state in bytes, for models with linear-attention layers.

Model and hardware profiles. All constants above follow from a model profile, a GPU profile, and two number formats: the weight format q_w and the activation format q_a. With $\beta(q)$ the bytes per element of format q ($\beta(\text{BF16}) = 2$, $\beta(\text{FP8}) = 1$, $\beta(\text{NVFP4}) = 0.5625$),

$$\begin{gathered} W = N_{\text{total}} \cdot \beta(q_w), \quad b_{\text{kv}} = e_{\text{kv}} \cdot \beta(q_a), \quad A = N_{\text{active}}, \\ \text{FLOP/s} = F_{\text{GPU}}(q_c), \quad \text{BW} = \text{BW}_{\text{GPU}}, \end{gathered} \tag{7}$$

where N_{total} and N_{active} are the total and active parameter counts, e_{kv} the number of KV cache elements per token, and $F_{\text{GPU}}(q)$ the peak dense tensor-core throughput of the GPU in format q.

Retrieval backend latency. Additionally, the total latency of a request adds a fixed delay for every turn that queries the retrieval backend:

$$L = \sum_{p} t(p) + \tau_{\text{backend}} \cdot N_{\text{backend}}, \tag{8}$$

where $\tau_{\text{backend}} = 0.1\,\text{s}$ and N_{backend} is the number of turns with at least one retrieval call. All retrieval calls issued in the same turn count as one round, making parallel tool calls cheap and sequential tool calls expensive.

Future versions of this latency model could account for query complexity. SID-2 learned to construct complex filters and use many BM25 terms, which substantially increased backend latency beyond the fixed delay assumed here. We estimate backend latency rather than measuring it directly, because the high request concurrency during training produces observed latencies that are unrepresentative of what a user would see in production.

3.4. Proposer-Solver Self Play

Scaling RL on LLMs is limited by compute, training stability, and the availability of appropriately difficult tasks with known solutions. Search is a uniquely attractive candidate for eliminating the third bottleneck, because every task is a query that can be answered by searching a corpus, and a sufficiently large corpus provides a practically limitless supply of potential answers. SID-2 is our first model that moves in this direction. Specifically, we let it propose queries that it then attempts to answer by searching the corpus. Proposer episodes are rewarded for queries that provide useful training signal, whereas solver episodes are rewarded with our standard search rewards.

While proposer/solver episodes do not comprise all of our training data, they do confer several desirable properties even at this scale:

1. **Data-free scaling.** We can improve search performance on any corpus without the need to manually generate or buy additional queries.
2. **Less jaggedness.** Despite performing well on harder queries, SID-1 occasionally struggled with simpler ones. The classic solution to this "jaggedness" is adding easy queries to the training mix. However, doing so indiscriminately creates groups where every attempt in the group scores perfectly, so every advantage is zero and the group contributes no training signal. Oversampling can mitigate this, but it cannot resolve it entirely (Yu et al., 2025). In proposer/solver training, the model is incentivized to exploit these regressions as they appear, creating an online curriculum.

Because the proposer and the solver share weights, our early attempts suffered from "subliminal" communication, in which the model proposed a query paired with a seemingly unrelated set of golden documents, but the solver inexplicably found many of these spurious documents. We do not fully investigate and ablate the root causes, but we posit that it is related to the effects described in (Cloud et al., 2025). We managed to address these effects without adjusting the policy gradient or loss function.

We find self-play to be particularly effective for "needle-in-a-haystack"-style queries, whereby there is a single golden document to find which is easy to verify as correct once found. For a given proposed query, we reward the proposer by performing n episodes with the solver and computing:

$$R_{\text{proposer}} = n_{\text{NDCG=0}} \times n_{\text{NDCG=1}} \tag{9}$$

where:

- $n_{\text{NDCG=0}}$ is the number of solver episodes that had an NDCG of 0 (i.e., did not report the golden document at all) and
- $n_{\text{NDCG=1}}$ is the number of solver episodes that had an NDCG of 1 (i.e., found the golden document and ranked it first)

We plot this proposer reward across a training run with a frozen SID-1 solver in Figure 7. By having the solver provide a ranked list of possible solutions and only having the proposer be rewarded if the answer is ranked first in some episodes but completely omitted in others, we circumvent the problem of the proposer being rewarded for vagueness noted by Brown (2025).

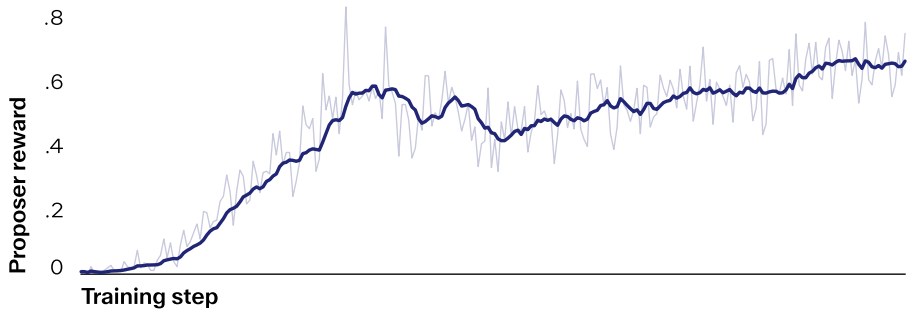

Figure 7 | Proposer reward R_{proposer} over training for a proposer run with a frozen SID-1 solver. The faint line is the per-step reward; the solid line is a 10-step rolling mean.

Related work. In classical RL domains, this kind of "self-play" has tended to outperform models that rely entirely on human-generated training examples. For example, AlphaZero plays chess against itself and routinely beats chess engines trained on human games alone (Silver et al., 2018). We believe removing human priors from search model training will similarly unlock new performance levels.

3.5. Harness

SID-2 is trained with multi-turn RL in a retrieval harness. We also publish the harness. During each episode, it searches at least one corpus, reads documents, and reports a ranked list of document IDs once it has enough context to answer the query or is confident that sufficient context does not exist. Each search result and each passage read lands in the model's context window. Because context is limited and additional context increases latency, we designed the harness to teach SID-2 to manage its context efficiently without sacrificing result quality: SID-2 decides how much text it sees and never sees the same text twice. This allows SID-2 to search documents of arbitrary length and corpora of arbitrary size.

Search results fed back to SID-2 contain only a relevant snippet from each document. The model only reads a document in its entirety when it judges that doing so would be helpful, and as a result uses the read tool infrequently. To select each snippet, we slide a fixed-size window over the document, score each position against the query with BM25 over unigrams and bigrams, and return only the highest-scoring window. The model controls both the number of snippets returned (`limit`) and the size of each snippet (`snippet_size`). Neither is directly rewarded or penalized; more and larger snippets cost the model only through the latency terms of the reward. The model can use the read tool to expand the current snippet, or request a larger snippet on the next search.

A potential risk with iterative search is context pollution from redundant reads, particularly when refined queries return overlapping results. To prevent this, each snippet carries a reference `id#start:end` that denotes the character range shown. The harness caches every document surfaced in an episode and keeps track of the character ranges that the model has already seen. If a subsequent search returns redundant text, the redundant range is replaced with a marker, such as `[seen: "#120:480"]`. Reads are served from the cache and require no additional round trip to the backend. The model also never sees database IDs. Instead, each document receives a new ID when it first appears in an episode, preventing the policy from memorizing document IDs across epochs.

The same policy must serve real-world deployments that differ in corpus type, search backend and retrieval objective. To ensure that SID-2 generalizes, we vary all three during training. First, we train on a wide range of corpora spanning science, medicine, finance, email, law, web, customer support and structured records. Some environments include more than one of these corpus types, each exposed through its own prefixed tools. Second, we vary tool availability both across and within corpora, because we expect some SID-2 deployments to have only a lexical or only a semantic index, while others will have both. Third, each training row samples its objective (NDCG, F1, or Recall@k for $k \in \{1, 2, 5, 10, 20, 50\}$), so a single policy learns to trade precision for recall (or vice versa) when prompted. Conveniently, the F1 and small-k Recall@k objectives also act as a "fast" mode, so SID-2 can prioritize speed, exhaustiveness, or precision depending on the stated retrieval objective.

3.6. Data

In SID-1, we found that public benchmarks such as HotpotQA (Yang et al., 2018) are largely saturated. For SID-2, we expand both the search spaces and the kinds of tasks during training. We add the following task types to cover a wider range of search use cases:

- **Multi-corpus.** Relevant documents are spread across several corpora. The model must decide where to search and combine results across collections.
- **Filtered.** Queries include constraints, such as inclusions and exclusions, that map to metadata filters on the corpus.
- **Tabular data.** The information needed to answer a query is stored in table rows, requiring the model to interpret fields and relationships between them.
- **Findall.** Queries ask for every document satisfying a set of constraints, for example every contract containing a given clause. These tasks require the model to keep searching after finding a relevant document and decide when the result set is complete.

We specifically remove riddle-like questions. Riddle questions are superficially attractive, because they are difficult and provide the intra-group reward differences required for group-relative training. This makes them common in search benchmarks and synthetic datasets, for example Browsecomp-Plus.

But riddle-like questions are not realistic: Outside of game-shows, few people ask unnecessarily obfuscated queries. For example, riddle-like questions replace common terms with indirect descriptions and or rare synonyms (the "18000 island nation" instead of "Indonesia"). We believe that optimizing puzzle solving skews performance away from the searches people and agents actually perform. One particular such example is Browsecomp-Plus.

4. SID-2 Swarm

Historically, scaling parameter count has been the primary driver of AI progress (Hoffmann et al., 2022; Kaplan et al., 2020). More recently, two other levers have risen in popularity, namely scaling reasoning (DeepSeek-AI, 2025; OpenAI, 2024) and scaling the number of agents (Kimi Team, 2026a; OpenAI, 2026b). Unlike in mathematics and coding, we find that scaling the number of agents appears to be a particularly effective driver of search performance.

In our SID-1 technical report (SID Research, 2025), we found that fusing the search results from four uncoordinated agents tended to beat those of a single agent on recall and NDCG. We believe this is because each takes a slightly different search strategy. As such, we highlighted forked contexts as an important direction for future work. SID-2 Swarm pursues this coordinated division of labor.

In Figure 8 we compare Reciprocal Rank Fusion (RRF) of isolated SID-2 runs with SID-2 Swarm, trained in our multi-agent harness. We observe that scaling agents in the swarm has a higher return in FLOPs to downstream retrieval performance than scaling isolated agents.

In this section, we explain our thesis on why scaling the number of agents is so effective in search (§4.1), our process for training SID-2 Swarm (§§4.2–4.5), and promising future directions for search agent swarms (§4.6).

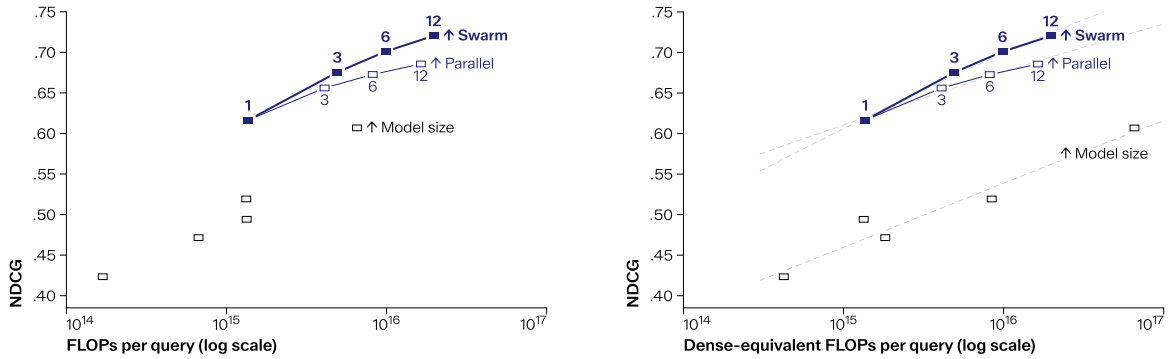

Figure 8 | NDCG against compute per query for Reciprocal Rank Fusion (RRF) of 3, 6 and 12 independent SID-2 runs (parallel) and for SID-2 Swarm with 3, 6 and 12 agents (swarm), both starting from a single SID-2 run. The points on the two lines are labeled with the number of agents; the other points are open-weight base models of different sizes, each run as a single agent (model size). Compute is estimated as 2× parameters × the context tokens of all agents. *Left:* actual compute, counting the active parameters of mixture-of-experts (MoE) models. *Right:* dense-equivalent compute, where for MoE models we use the size of the dense model with the same predicted loss under the fine-grained MoE scaling laws of Krajewski et al. (2024), assuming 36T training tokens, with NDCG on a logit scale, log(NDCG/(1 – NDCG)), and ticks labeled in NDCG. Dashed lines are least-squares fits of logit(NDCG) against log compute for each series.

4.1. Search parallelizes well

Search is a divisible task. Recently, Ord (2026) found that web search was much more parallelizable than other tasks for agent swarms. Specifically, a 2× increase in parallel tokens yielded similar performance to an ≈ 1.72× increase in sequential tokens for web search, in comparison to an ≈ 1.48× increase in finding software vulnerabilities and an ≈ 1.39× increase in general terminal tasks.

Search algorithms have a tradeoff between breadth (inspecting many items) and depth (inspecting each item more deeply). We hypothesize that for many natural language search use cases there is a high proposer-verifier gap, so that scaling depth saturates early, meaning that scaling breadth is often more advantageous.

In the case of agentic search, the breadth vs. depth tradeoff manifests as a design choice between using many small models in parallel (which will look at more documents) or a single large model (which will expend more FLOPs per document). Provided a document, determining whether it answers a query is usually straightforward, so we hypothesize that swarms of small models will beat fewer, larger models on most search tasks.

Scaling to parallel agents also has attractive properties for decreasing latency: While single-agent generation is bounded by the tokens per second per user (i.e., interactivity), multiple agents can be scaled arbitrarily using concurrent generation within and between replicas. Further, within a deployed replica, forking agents on a shared prefix can ease total memory (Kwon et al., 2023; Zheng et al., 2024) and memory bandwidth (Juravsky et al., 2024) consumption by the KV cache.

4.2. Multi-agent harness design

Our initial experiments in this direction used a harness with separate roles of *stem* and *fork* agents. *Stem* agents could only call a `fork` tool, which would spawn *fork* agents that share their context with the stem. *Fork* agents can search and push helpful documents back to the stem. This harness was hand-designed to induce a large amount of KV-cache re-use and ensure minimize context pollution for the stem.

Unfortunately, we observed models would (1) persistently be confused as to their role and try to call tools that aren't available to them and (2) the stem model would have a hard time understanding how the search tools would behave (as it could no longer learn this in-context).

We gradually relaxed the harness we used, eventually converging on a harness similar to OpenAI's multi-agent harness (OpenAI, 2026b). In this harness, every agent can call:

- the search tools;
- `fork`: spawns new agents with specified names and forked context (the maximum number of agents is capped globally);
- `message`: sends a textual message to a list of agents' inboxes;
- `wait`: waits until a message arrives in this agent's inbox;
- `list`: lists all the agents and their status;
- `report_helpful_ids`: ends the calling agent's part of the episode and reports its most helpful documents by passing their identifiers.

Messages in a given agent's inbox are appended to the result of its next tool call. If a message is not delivered (because, for example, the recipient has already reported), the sender receives textual feedback. At the end of the episode, all the documents reported across `report_helpful_ids` calls are fused with RRF.

For both harnesses, we compute the score of the whole episode at its end and apply the resulting advantage evenly to all agents in the episode. By viewing the other agents' messages as observations from the RL environment, we can see that the standard policy gradient theorem generalizes to this multi-agent setting.

4.3. Simulated-latency request scheduling

To maximize throughput during RL training, generations are generally configured to be asynchronously scheduled by environments. For our single-agent setup, the wall-clock time at which requests are scheduled does not affect the environment observations or reward, and so does not materially affect the training process. However, for multi-agent training, scheduling affects the order in which messages from other agents arrive, letting agents become arbitrarily out of sync with one another. This can become particularly egregious when agents in the same episode are routed to different rollout engines, some of which can have a long request queue. As deployment is configured for lower throughput to decrease latency, we expect agents to be largely in sync, creating a mismatch between the behavior we encourage in training and the one we desire in deployment. We illustrate this effect in Figure 9.

To remedy this, we leverage our *Latency Model* (§3.3) to attach a simulated latency to every completion and message. If one agent is more than two seconds of simulated latency ahead of others then the harness will wait for other agents to catch up before scheduling its next turn. Likewise, a message whose simulated send time is ahead of its recipient's clock is held back until the recipient catches up.

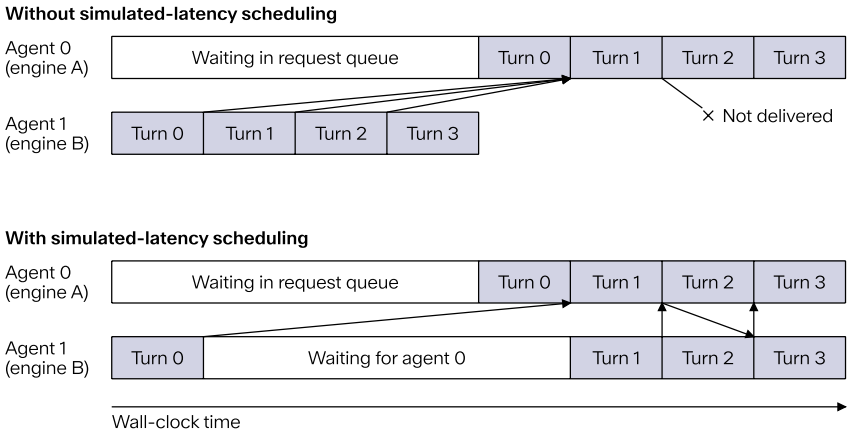

Figure 9 | Simulated latency allows messages to be delivered closer to their ordering in deployment. With no simulated latency scheduling [top], agent 1 can complete its whole episode while agent 0 waits in the request queue for a highly contested rollout engine. When simulated latency scheduling is applied [bottom], the turn 1 completion for agent 1 waits for agent 0 to finish its first completion, meaning that agents can tightly coordinate their strategies. Arrows are messages, delivered at the start of the recipient's next turn.

4.4. Stabilizing entropy for exploration

We found messaging behavior during training to be unstable. In early experimental training runs, it often collapsed to local minima in the policy space, whereby all the agents worked almost independently without communication, leading to performance approximately equivalent to RRF. Once set in, this behavior was permanent, and communication did not recover with further training.

One apparent driving factor is the policy entropy: We anecdotally observe fewer messages when entropy decreases. To test whether entropy drops are *causing* the messaging behavior to collapse, we select a single checkpoint and sweep the temperature from 0 to 1. In Figure 10 (left) we plot our results, showing a clear relationship between forcing lower entropy by reducing the

sampling temperature and a lower number of messages sent by agents. We hypothesize that the causal mechanism is the base model's lack of training in an agent-swarm setting, which makes using the collaborative tools an exploratory action.

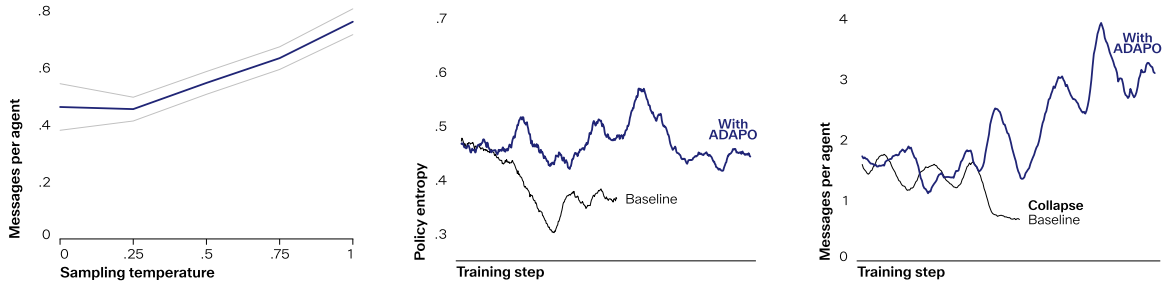

Figure 10 | Entropy drives messaging behavior. [Left] Mean number of messages sent per agent when the same checkpoint is sampled at temperatures from 0 to 1, over 40 queries (one sample per query at temperature 0, three at the others). The thin lines show ±1 standard error over episodes. [Middle and right] Policy entropy (middle) and mean messages sent per agent (right) over RL training, for a fixed $\epsilon_{\text{high}} = 0.26$ baseline and for the same run continued from a checkpoint at step 255 with ADAPO, target entropy 0.5. The lines are rolling means.

To remedy this entropy-driven collaboration collapse in live training runs, we consider the PPO upper clipping parameter, ϵ_{high}. This parameter was shown by Yu et al. (2025) to steer entropy by affecting the rate at which low probability tokens are promoted. We find that setting this clipping parameter to a fixed value is insufficient for a stable entropy throughout the training run, and so use ADAPO (Petrenko et al., 2026), whereby this parameter is dynamically adjusted up or down based on a target entropy. In Figure 10 (middle and right) we demonstrate how introducing ADAPO with a target entropy of 0.5 leads to a more stable entropy and steadily growing collaborative behavior throughout the run compared to a baseline with a fixed $\epsilon_{\text{high}} = 0.26$.

4.5. Emergent collaborative behaviors

As shown in Figure 10 (right), with sufficient exploration agents learn to engage in more collaborative behavior throughout the training run. We study these learned collaborative behaviors to understand where the gains over naive RRF (i.e., isolated agents that send no messages) come from.

We display results for an example query from a held-out legal corpus for 3 agents in Figure 11 and for 12 agents in Appendix A.1. We observe that agents tend to reciprocate messages, ask for more information or delegate finding information, and send potential golden documents to each other. Interestingly, we do not observe an emergence of "management structures" or coordinator agents. We hypothesize that this is due to the high proposer-verifier gap in search making information bottlenecks, and management structures to support them, less salient, as it often suffices for agents to send a list of interesting documents.

We notice that agents spend much less effort on convincing each other of ideas and potential leads than human teams. We believe this is because all agents in an episode receive the same reward and are thus incentivized to collaborate and trust each other more than human teams.

QUERY

In a California civil case, an employer answered a complaint, paid jury fees, served and responded to written discovery, participated in meet-and-confer efforts, and took the plaintiff's deposition before moving to compel arbitration about a year after suit was filed. Under California law as it stood before Morgan v. Sundance, what legal standard governs whether the employer waived the right to compel arbitration, and is mere participation in litigation plus litigation expense enough to show waiver?

`root` `fork` → `arbitration-waiver-standard`, `arbitration-waiver-elements`

`arbitration-waiver-standard` → `arbitration-waiver-elements`

Key findings so far. The governing standard in California (pre-Morgan) is from St. Agnes Medical Center v. PacifiCare of California (2003) 31 Cal.4th 1187, which quotes the 6 factors from Sobremonte. The key language: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses." (St. Agnes at p.1203, citing Groom v. Health Net). Also very relevant: Keating v. Superior Court (1982) 31 Cal.3d 584 at 605: "waiver does not occur by mere participation in litigation." The question asks whether mere litigation participation + expense is enough — the answer is NO, you need more than that (e.g., use of judicial discovery to gain strategy info, or actual litigation of the merits). Can you find additional California cases with facts matching employer answered, paid jury fees, served discovery, meet & confer, deposition, then motion to compel ~1 year later?

`arbitration-waiver-elements` → `arbitration-waiver-standard`

Key findings so far. The key California cases are: 1) Saint Agnes Medical Center v. PacifiCare (2003) - states the 6-factor test and says "courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses." 2) Davis v. Continental Airlines (1997) - CA Court of Appeal, discusses employer doing extensive discovery (1,600 pages, 2-day deposition) before motion to compel arbitration - found waiver. 3) Keating v. Superior Court (1982) - "waiver does not occur by mere participation in litigation; some prejudice must be shown." 4) Christensen v. Dewor Developments (1983) - "no single test; factors: (1) steps inconsistent with intent to arbitrate, (2) unreasonable delay, (3) bad faith/wilful misconduct." Can you search further for cases with the specific fact pattern: employer answered complaint, paid jury fees, served/written discovery, meet-and-confer, deposition, moved to compel ~1 year after suit filed?

`arbitration-waiver-standard` → `arbitration-waiver-elements`

Thanks for the info. I'm searching for the specific case with these exact facts. Can you search for California cases where the opinion mentions the phrase "paid" in proximity to "jury fees" as a fact in the procedural history, in the context of an arbitration waiver analysis? Or perhaps cases with phrase "served and responded" in relation to written discovery in an arbitration context?

`root` → `arbitration-waiver-standard`, `arbitration-waiver-elements`

Key findings so far. The most relevant chunk IDs to report are:

1. St. Agnes Medical Center v. PacifiCare of California (Cal. Supreme Court) - key chunks: • 3y7idFFZtdJ - six-factor test for waiver • JNMHvR - "waiver does not occur by mere participation in litigation" + "courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses" • SU3Q - where prejudice typically IS found • ud8mJT7O - beginning of prejudice discussion

2. Roman v. Superior Court (Cal. App. 4th, 2009) - key chunks: • ezIJ - six factors • 56WAO - "mere participation" and "litigation expenses are insufficient"

3. Groom v. Health Net (Cal. App., 2000): • OkIds9qCRx - discusses the standard

4. Harvey v. Yellowpages.Com (unpublished): • 3PcOrk1v3aZ - repeats Roman's language

5. Oregel v. PacPizza (published) - discusses jury fees in context of waiver

The answer to the question: Under pre-Morgan California law, the standard is "actual prejudice." Mere participation in litigation plus litigation expense (court costs and legal fees) is NOT enough to constitute actual prejudice. The key holding from St. Agnes is that courts "will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses."

`arbitration-waiver-elements` → `arbitration-waiver-standard`

I'm about to call report_helpful_ids. Key chunk IDs to report in order of relevance: 1. JNMHvR (Saint Agnes, "courts will not find prejudice where only court costs and legal expenses") 2. 3y7idFFZtdJ (Saint Agnes, six-factor test from Sobremonte) 3. WfHvZRSb3b (Christensen, three-factor test quoted from Keating) 4. W2663UWe8SxG (Keating, same factor list + "waiver could occur before judgment on merits if prejudice shown") 5. t6Mfz4QK (Davis v. Continental, applies standard to employer/employee context) 6. 56WAO (Roman, "those expenses are insufficient by themselves") 7. xFmWR2Cfrct (Doers, "judicial litigation of the merits of arbitrable issues") 8. HNSnBipA (Davis v. Continental, detailed analysis of prejudice) 9. 9EmVqs (Davis, more detail on Christensen) 10. c8kTFRSyh24P (Keating, similar to W2663UWe8SxG) 11. mF1WiqV (Keating) 12. rlMay7YIF (Saint Agnes, follow-on analysis) 13. 3y7idFFZtdJ covered 14. QNtqxUyoetRn (Engalla, waiver "not lightly inferred") 15. 3Ba7uFVo (Oregel, jury fees context) Have you found any additional chunk IDs that should be included?

Figure 11 | Messages between the agents of one SID-2 Swarm episode on a held-out legal query (three-agent cap). The root forks two sub-agents, which exchange the authorities they have found and ask each other to search for cases matching the query's facts; before reporting, the root and a sub-agent each share a ranked list of chunk ids. Searches, reads and `wait` calls are omitted. Appendix A.1 gives the full transcript for the same query with a twelve-agent cap.

4.6. Future work

Given the promising returns to scaling the number of parallel agents in search, we consider the following avenues as promising directions for aiding in scaling to 100s or 1000s of parallel agents:

More flexible cooperation tools. Currently, messaging a list of 100 agents would require putting all these agents' names into the tool parameter. While in theory feasible, in practice this is quite unwieldy. Allowing the creation of "agent groups", for instance Zarov et al. (2026)'s Slack-like message board is a compelling approach. We believe that even more flexible harnesses are possible with models given access to a code interpreter for message sending.

Credit assignment. As other authors have noted (Hou et al., 2026; rednote hi lab, 2026; Z.ai, 2026), variance of the GRPO policy gradient grows with the number of tokens in the episode for single-agent RL, making value models advantageous in long-context settings. A similar variance growth is true for scaling the number of agents, but the selection of what information to expose to the value model in a multi-agent setting is much less clear.

Low latency agent swarms. Experiments in this section focused on the high reasoning effort regime. We hypothesize that there exist very effective strategies for agent swarms at lower reasoning efforts, for example having agents delay calling the `fork` tool until they are confident that they are performing better than other agents in the same episode.

5. Infrastructure

5.1. Training optimizations

We dedicate significant effort to increasing our training pipeline throughput. We use a fully asynchronous training pipeline using PipelineRL (Piché et al.) with an EAGLE draft head for rollout (Li et al., 2024); in serving, we use DFlash (Chen et al., 2026) instead (*Latency Model*, §3.3). Leveraging roofline throughput estimates assuming good memory bandwidth and FLOPs utilization, we carefully analyze cross-device communication bottlenecks to configure expert, tensor, data and context parallelism. We also control how many episodes are scheduled in parallel to mitigate re-prefilling tokens during rollout. In total, these optimizations compounded to achieve a 6× boost in training throughput over a naive approach using SLIME's default implementation.

5.2. Train-inference mismatch

We take care to mitigate two types of mismatch throughout training, rollout and deployment: (1) tokenization mismatches (SID Research, 2025) that cause catastrophic collapse of models during training and invalidate the KV cache in deployment, increasing costs and latency, and (2) numerical mismatches between training and inference that degrade performance (He and Thinking Machines Lab, 2025; Yao et al., 2025).

5.2.1. Responses API for eliminating tokenization mismatches

To avoid tokenization mismatches, we design our training framework to be fully Tokens-In/Tokens-Out (TI/TO). This completely eliminates tokenization mismatches during RL runs (SID Research, 2025), but naively means that RL environments must be written in tokens, presenting difficulties when evaluating models with different tokenizers or that are only avail-

able via API. To respect TI/TO while writing environments in a messages-based format, we base our training infrastructure on the Responses API (OpenAI, 2025) via a *Responses Translation Layer*, which we keep separate from the rollout engine. During an RL run, in `store=true` mode this translation layer presents a Responses-compatible endpoint to the environments, but it also stores the exact input and output tokens in a given turn and chains turns together with the `previous_response_id` parameter. For multi-agent training (§4), the `previous_response_id`s form a tree. For each new request during training, the translation layer reconstructs the exact chain of preceding tokens via tracing this tree, renders the new message in tokens and parses the model response back to a message. When an episode is finished, this chain of tokens is extracted and passed to the training engine.

To respect Zero Data Retention (ZDR) during deployment, the Responses Translation Layer encrypts the preceding tokens and sends them to the user; in the next turn, the user returns these encrypted tokens with the new message. This allows us to share tool call parsers and message renderers between training and deployment (eliminating the risk of performance-degrading mismatches) and guarantee that KV cache is not missed due to tokenization mismatch, boosting KV cache hit rate from 50% to 80% on our first party API.

5.2.2. Importance sampling and infrastructural mitigations for numerical mismatch

Even with the exact same tokens, numerical mismatches between training and inference engines can cause subtle differences between the rollout policy $\pi_\theta^{\texttt{SGLang}}$ and the training policy $\pi_\theta^{\texttt{Megatron}}$.

We leverage token-level Truncated Importance Sampling (TIS) (Yao et al., 2025), which gives the theoretical guarantee that for a single generated token (Zheng et al., 2025a), the policy gradient is unbiased. Specifically, where A is the advantage, $\mathcal{J}_\theta^{\texttt{Megatron}}$ is the expected reward under the Megatron policy and x is a single token:

$$\begin{aligned}
\nabla_\theta \mathcal{J}_\theta^{\texttt{Megatron}} &= \nabla_\theta \mathbb{E}_{x\sim\pi_\theta^{\texttt{Megatron}}}(A) \\
&= \nabla_\theta \mathbb{E}_{x\sim\pi_{\theta^*}^{\texttt{SGLang}}}\left(\frac{\pi_\theta^{\texttt{Megatron}}(x)}{\pi_{\theta^*}^{\texttt{SGLang}}(x)}A\right) \\
&= \mathbb{E}_{x\sim\pi_{\theta^*}^{\texttt{SGLang}}}\left(\frac{\nabla_\theta\pi_\theta^{\texttt{Megatron}}(x)}{\pi_{\theta^*}^{\texttt{SGLang}}(x)}A\right)
\end{aligned}$$

Where θ^* represents frozen (i.e., non-differentiated) weights. However, there is *no guarantee* that increasing $\mathcal{J}_\theta^{\texttt{Megatron}}$ with a sufficiently small gradient step will increase the expected reward under the rollout policy $\mathcal{J}_\theta^{\texttt{SGLang}}$. In fact, as shown previously by Ma et al. (2026b), entirely different experts, and thus neural circuitry, can be activated during rollout and training: The circuits reinforced during training may be different from the circuits that produced the training sample.

We take infrastructural steps to align the rollout and training policies closely. We use Rollout Routing Replay (R3) (Ma et al., 2026b) to reduce the train-inference mismatch. R3 delivers a 3-fold reduction in the absolute difference between the log probabilities, $|\log p_{\text{train}}(x) - \log p_{\text{inference}}(x)|$. In doing so, we discovered a bug whereby the router in the EAGLE head used for inference would overwrite the R3 information for the target model in the open source SGLang implementation, leading to the layer-0 replayed experts being the indices from the draft model rather than

the target model. In Figure 12, we observe that fixing this bug leads to a significant improvement in the log-probability difference and the downstream validation performance. This bug has since been fixed by other contributors (Li, 2026).

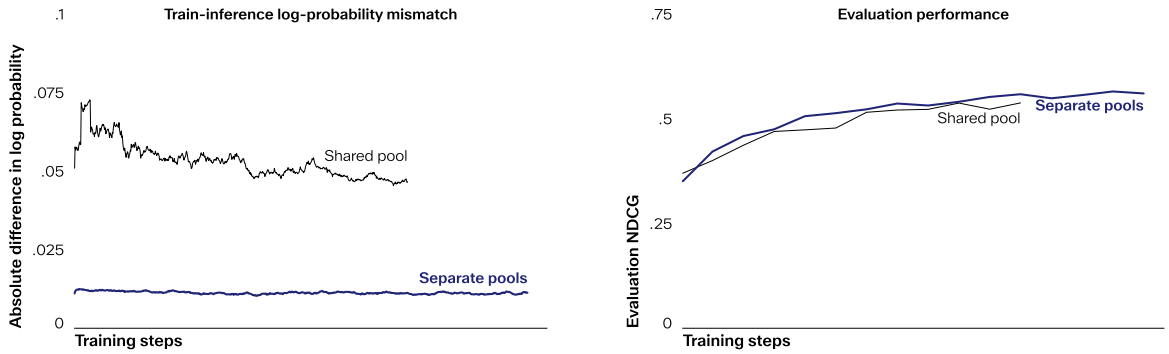

Figure 12 | Train-inference mismatch, measured as $|\log p_{\text{train}}(x) - \log p_{\text{inference}}(x)|$ (left) and evaluation NDCG (right) when the EAGLE draft head shares the target model's R3 expert pool (the bug) and when it has its own. Runs shown here were part of the initial experimentation in the development of SID-2.

Prior work has found that computing the LM head and the following logits in FP32 reduces the log-probability difference (MiniMax, 2025). We find that it is sufficient to keep the LM head in BF16 and upcast its output logits to FP32 before the softmax calculation, allowing us to still use the higher-throughput BF16 tensor cores for this matmul.

6. Results

Tables 2–4 of Appendix B summarize NDCG, cost and estimated latency performance by model on our evaluation mix. SID-2 Swarm with 12 agents achieves the highest aggregate NDCG, and GPT-6 Astra at maximum reasoning effort comes in second place. Our evaluation mix consists of 506 search questions across 19 corpora. It includes 207 public benchmark questions and 299 questions generated by our proposer. We observe meaningful variation in performance across models and corpora. While copious ink could be spilled decomposing relative performance, we briefly highlight four major themes.

First, some search tasks can be tackled effectively by a single agent using a small model. For example, on the WixQA customer support benchmark, a single SID-2 agent at high reasoning matches GPT-6 Astra at maximum reasoning on NDCG but at approximately 232× lower cost and 27× lower estimated inference latency. Similarly, we observe little NDCG benefit from unleashing Swarm on WixQA. These results are particularly relevant to latency-sensitive applications where straightforward questions are common, such as live chat.

Second, several corpora in particular benefit from swarms. SID-2 Swarm with 12 agents leads on Enron emails, EnterpriseRAG-Bench, LinkedIn profiles, and the directory of Y Combinator companies. On EnterpriseRAG-Bench, for example, SID-2 Swarm with 3 agents takes second place, slightly above GPT-6 Astra. Given that this is a synthetic corpus, we hypothesize that swarms are particularly effective when models have no prior knowledge of a corpus.

Third, the complex and large legal, financial and scientific corpora present a more mixed bag. Astra leads on California law, SEC filings, and FiQA, while Swarm leads on BarExamQA, PubMed and arXiv. We hypothesize that Astra's prior knowledge of California case law and public companies helps it locate relevant evidence more easily, although reasoning ability may

be another factor. Scaling Swarm beyond 12 agents to match Astra's compute budget would isolate the effect of reasoning.

Fourth, completion reliability greatly affects performance. Inkling, Nemotron, Beam, and GLM models frequently exhaust the 10-turn limit without reporting results, despite instructions not to do so. Opus 5.5 refused to answer 30 of the 43 SciFact questions, causing it to perform poorly on this benchmark. We discuss these idiosyncrasies and our methodology in greater detail in Appendix B.

Taken together, the results in Tables 2–4 support distinct use cases for SID-2 and Swarm. For straightforward search tasks, SID-2 provides low-cost, low-latency retrieval at rough NDCG parity, while Swarm reliably delivers higher NDCG where additional effort is needed. Astra retains a modest NDCG advantage on several legal and financial benchmarks but at a significant latency and financial cost. Given the small sample sizes for individual corpora, evaluating models on representative production data remains critical.

7. Conclusion

Building on SID Research (2025), which found that agentic search improves retrieval performance relative to fixed pipelines, we have shown that coordinated search agent swarms achieve higher retrieval quality within a given compute budget than a single Astra-class agent or multiple uncoordinated agents whose results are fused post hoc. We have also shown that configurable reasoning effort and Metric Steering can help tailor agentic search to application requirements. In future work, we plan to explore more efficient swarms through better cooperation tools and credit assignment, lower-latency coordination, and proposer–solver training on increasingly realistic tasks. As with past research, the goal of our future research will be to build better tools for knowledge work agents to find and build on prior work.

8. Citation

Please cite this work as:

> SID Research, "SID-2 Technical Report", SID AI, Oct 2026.

Or use the BibTeX citation:

```
@article{sid2026sid2,
  author  = {SID Research},
  title   = {SID-2 Technical Report},
  journal = {SID AI},
  year    = {2026},
}
```

References

N. Brown. Why self play works for two-player zero-sum games but is so much harder to use in "real world" domains. Thread on X, Oct. 2025. URL https://x.com/polynoamial/status/1980697004658556972. Posted 21 October 2025.

Chen et al. DFlash: Block diffusion for flash speculative decoding, 2026.

Z. Chen, X. Ma, S. Zhuang, P. Nie, K. Zou, A. Liu, J. Green, K. Patel, R. Meng, M. Su, S. Sharifymoghaddam, Y. Li, H. Hong, X. Shi, X. Liu, N. Thakur, C. Zhang, L. Gao, W. Chen, and J. Lin. BrowseComp-Plus: A more fair and transparent evaluation benchmark of deep-research agent, 2025. URL https://arxiv.org/abs/2508.06600.

A. Cloud, M. Le, J. Chua, J. Betley, A. Sztyber-Betley, J. Hilton, S. Marks, and O. Evans. Subliminal learning: Language models transmit behavioral traits via hidden signals in data, 2025. URL https://alignment.anthropic.com/2025/subliminal-learning/.

Cognition. Introducing SWE-2: Pushing the pareto frontier, 2026.

D. Cohen, L. Burg, S. Pykhnivskyi, H. Gur, S. Kovynov, O. Atzmon, and G. Barkan. WixQA: A multi-dataset benchmark for enterprise retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2505.08643.

K. Corbitt. ART·E: How we built an email research agent that beats o3. OpenPipe blog, 2025. URL https://openpipe.ai/blog/art-e-mail-agent. TODO: check author and title.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI. DeepSeek-V4 technical report. Technical report, 2026.

L. Emberson and D. Roodman. The plunging price of thought. Published online at epoch.ai, 2026. URL https://epoch.ai/publications/the-plunging-price-of-thought. Accessed 25 Oct 2026.

H. He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250910. https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/.

J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30031, 2022. URL https://arxiv.org/abs/2203.15556.

Z. Hou, Y. Li, J. Tang, and Y. Dong. Single-rollout asynchronous optimization for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2607.07508.

G. Ilharco et al. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

J. Juravsky, B. Brown, R. Ehrlich, D. Y. Fu, C. Ré, and A. Mirhoseini. Hydragen: High-throughput LLM inference with shared prefixes, 2024.

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

A. Kharazian. Ramp AI index. Ramp, 2026. URL https://ramp.com/data/ai-index?metric=token-volume&detail=maker&mode=volume. Token-volume series. Accessed 26 Oct 2026.

Kimi Team. Kimi K2.5: Visual agentic intelligence, 2026a. URL https://arxiv.org/abs/2602.02276. Introduces Agent Swarm and Parallel-Agent Reinforcement Learning (PARL).

Kimi Team. Kimi K3: Open frontier intelligence. Technical report, 2026b.

J. Krajewski, J. Ludziejewski, K. Adamczewski, M. Pióro, M. Krutul, S. Antoniak, K. Ciebiera, K. Król, T. Odrzygóźdź, P. Sankowski, M. Cygan, and S. Jaszczur. Scaling laws for fine-grained mixture of experts. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.07871.

S. Krishna, K. Krishna, A. Mohananey, S. Schwarcz, A. Stambler, S. Upadhyay, and M. Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2409.12941.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pages 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

J. Li. [fix] Skip routed expert capture for draft model under spec v2. SGLang pull request #26980, 2026. URL https://github.com/sgl-project/sglang/pull/26980.

Y. Li, F. Wei, C. Zhang, and H. Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, 2024. URL https://proceedings.mlr.press/v235/li24bt.html.

Ma et al. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training, 2026a.

W. Ma, H. Zhang, L. Zhao, Y. Song, Y. Wang, F. Luo, and Z. Sui. Stabilizing MoE reinforcement learning by aligning training and inference routers. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://icml.cc/virtual/2026/poster/63177.

M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of The Web Conference 2018*, pages 1941–1942, 2018.

MiniMax. MiniMax-M1: Scaling test-time compute efficiently with lightning attention, 2025. URL https://arxiv.org/abs/2506.13585.

NVIDIA. NVIDIA nemotron 3 ultra technical report. Technical report, 2026.

OpenAI. Learning to reason with LLMs, Sept. 2024. URL https://openai.com/index/learning-to-reason-with-llms/. Announcement of OpenAI o1, 12 September 2024.

OpenAI. New tools for building agents, 2025. URL https://openai.com/index/new-tools-for-building-agents/.

OpenAI. Sharing AI progress in mathematics. Published online at openai.com, Oct. 2026a. URL https://openai.com/index/sharing-ai-progress-in-mathematics/. Accessed 26 Oct 2026.

OpenAI. Multi-agent, 2026b. URL https://developers.openai.com/api/docs/guides/responses-multi-agent. Responses API guide; released in beta on 9 July 2026 alongside GPT-5.6.

OpenAI. On the Navier–Stokes Millennium Prize problem. Published online at openai.com, Sept. 2026c. URL https://openai.com/index/navier-stokes-solution/. Accessed 26 Oct 2026.

T. Ord. Swarm scaling. LessWrong, 2026. URL https://www.lesswrong.com/posts/6cb7qd3RSkgnviCpf/swarm-scaling.

A. Petrenko, B. Lipkin, K. Chen, E. Wijmans, M. Cusumano-Towner, R. Giryes, and P. Krähenbühl. Entropy-preserving reinforcement learning. In *ICLR*, 2026. URL https://arxiv.org/abs/2603.11682.

A. Piché et al. PipelineRL. In *OpenReview*. URL https://openreview.net/forum?id=A35ak14Cyp.

rednote hi lab. TEMPO: Test-time-scaled value estimation with macro-step policy optimization. dots.ai blog, Aug. 2026. URL https://studio.dots.ai/dots/tempo-blog.html. Published 13 August 2026. Accessed 7 Nov 2026.

SID Research. SID-1 technical report: Test-time compute for retrieval. Technical report, SID AI, Dec. 2025. URL https://www.sid.ai/research/sid-1-technical-report.

D. Silver et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Y. Sun, J. Rahmfeld, C. Weaver, W. Chen, R. Desai, W. Huang, and M. H. Butler. EnterpriseRAG-Bench: A RAG benchmark for company internal knowledge, 2026. URL https://arxiv.org/abs/2605.05253.

R. S. Sutton. The bitter lesson. Published online at incompleteideas.net, 2019. URL http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

D. Tchuindjo, D. Shah, and O. Khattab. OBLIQ-Bench: Exposing overlooked bottlenecks in modern retrievers with latent and implicit queries, 2026. URL https://arxiv.org/abs/2605.06235.

N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7534–7550, 2020.

M. Wortsman et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

F. Yao, L. Liu, D. Zhang, C. Dong, J. Shang, and J. Gao. Your efficient RL framework secretly brings you off-policy RL training, Aug. 2025. URL https://fengyao.notion.site/off-policy-rl.

Q. Yu et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pages 125532–125554, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Z.ai. GLM-5.2: Built for long-horizon tasks. Z.ai blog, June 2026. URL https://z.ai/blog/glm-5.2. Published 16 June 2026. Accessed 7 Nov 2026.

I. Zarov, H. Course, A. Hartshorn, R. Achiron, L. Roche, A. Sinha, C. Taylor, and R. Taylor. Avalon-M1: Reinforcement learning from collective rewards, 2026. URL https://gr.inc/avalonm1.pdf.

C. Zheng, S. Liu, M. Li, X.-H. Chen, B. Yu, C. Gao, K. Dang, Y. Liu, R. Men, A. Yang, J. Zhou, and J. Lin. Group sequence policy optimization, 2025a. URL https://arxiv.org/abs/2507.18071.

L. Zheng, L. Yin, Z. Xie, C. Sun, J. Huang, C. H. Yu, S. Cao, C. Kozyrakis, I. Stoica, J. E. Gonzalez, C. Barrett, and Y. Sheng. SGLang: Efficient execution of structured language model programs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 62557–62583. Curran Associates, Inc., 2024. doi: 10.52202/079017-2000. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/724be4472168f31ba1c9ac630f15dec8-Paper-Conference.pdf.

L. Zheng, N. Guha, J. Arifov, S. Zhang, M. Skreta, C. D. Manning, P. Henderson, and D. E. Ho. A reasoning-focused legal retrieval benchmark. In *Proceedings of the 2025 Symposium on Computer Science and Law*, 2025b. doi: 10.1145/3709025.3712219. URL https://arxiv.org/abs/2505.03970.

A. Multi-agent coordination transcripts

Full coordination transcripts of SID-2 Swarm episodes, in the format of Figure 11: each `fork` and each message, as sender → recipients followed by the message text, in order of the simulated-latency clock. Searches, reads, `wait` calls and tool results are omitted.

A.1. Twelve agents on a legal query

The query of Figure 11, answered by the same checkpoint with a cap of twelve agents instead of three.

QUERY

In a California civil case, an employer answered a complaint, paid jury fees, served and responded to written discovery, participated in meet-and-confer efforts, and took the plaintiff's deposition before moving to compel arbitration about a year after suit was filed. Under California law as it stood before Morgan v. Sundance, what legal standard governs whether the employer waived the right to compel arbitration, and is mere participation in litigation plus litigation expense enough to show waiver?

`root` `fork` → `waiver-standard`, `waiver-cases`, `waiver-conduct`, `waiver-mere`, `waiver-expense`, `waiver-1281`, `waiver-SC`, `waiver-CA`, `waiver-morgan`, `waiver-history`, `waiver-participation`

`waiver-standard` → `waiver-conduct`, `waiver-mere`, `waiver-expense`

Key leads so far: St. Agnes Medical Center v. PacifiCare (2003) 31 Cal.4th 1187 is the key California Supreme Court case establishing the waiver standard. The factors from Sobremonte/Peterson are the analytical framework. The critical line is "the party seeking to establish a waiver bears a heavy burden of proof" and courts require conduct that "evinces an intent to waive the right to arbitrate." Cases like Roman v. Superior Court (2009) and the language "mere participation in litigation is not enough" are directly on point. Need to find published California cases specifically addressing the question of whether merely answering a complaint, paying jury fees, serving/answering discovery, meet-and-confer, and taking a deposition before moving to compel (with ~1 year delay) amounts to waiver.

`waiver-mere` → `waiver-standard`, `waiver-cases`, `waiver-conduct`, `waiver-mere`, `waiver-expense`, `waiver-1281`, `waiver-SC`, `waiver-CA`, `waiver-morgan`, `waiver-history`, `waiver-participation`

Key findings for the arbitration waiver question. The critical authority is St. Agnes Medical Center v. PacifiCare of California (2003) 31 Cal.4th 1187. Opinion ID 2519923 in the search system. Key chunks:

Chunk 6 (id 304893): States that "waivers are not to be lightly inferred and the party seeking to establish a waiver bears a heavy burden of proof."

Chunk 7 (id 151628): Contains the 6-factor test from Sobremonte/Peterson: (1) actions inconsistent with right to arbitrate; (2) litigation machinery substantially invoked and parties were well into preparation of lawsuit; (3) whether party requested arbitration close to trial date or delayed long period; (4) whether defendant filed counterclaim without asking for stay; (5) whether important intervening steps had taken place; (6) whether delay affected, misled, or prejudiced the opposing party.

Chunk 22 (id 275136): THE KEY LANGUAGE. States: "In California, whether or not litigation results in prejudice also is critical in waiver determinations." Then quotes Christensen: "Waiver does not occur by mere participation in litigation" if no judicial litigation of merits of arbitrable issues, but "waiver could occur prior to a judgment on the merits if prejudice could be demonstrated." Then: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses."

This answers the question: The standard is the 6-factor St. Agnes test with heavy burden on the party claiming waiver. Mere participation in litigation plus litigation expense is NOT enough to show waiver - there must be actual prejudice beyond mere costs/expenses.

Still need to find: (1) the Engalla opinion that first articulated the "preponderance of the evidence" standard (15 Cal.4th 948); (2) any cases specifically addressing the fact pattern in the question (employer answers complaint, pays jury fees, does discovery, takes deposition before moving to compel); (3) CCP section 1281.2.

`waiver-expense` → `waiver-standard`, `waiver-conduct`, `waiver-mere`

Found very relevant content. The key standard in California (pre-Morgan v. Sundance):
1. Saint Agnes v. PacifiCare (2003) 31 Cal.4th 1187 - Cal Supreme Court - lists the 6 factors and says "In California, whether or not litigation results in prejudice also is critical in waiver determinations"
2. Keating v. Superior Court (1982) 31 Cal.3d 584 - says "waiver does not occur by mere participation in litigation; there must be 'judicial litigation of the merits of arbitrable issues"'

3. Groom v. Health Net (2000) - says "Groom cites no case that holds that the mere expense of responding to preliminary court motions, by itself, constitutes the prejudice"

Question: Is there a specific California case that matches these exact facts (employer answered complaint, paid jury fees, served and responded to discovery, meet and confer, took deposition, then moved to compel about a year later)? Also look for the specific quotes/standard from Saint Agnes at pages 1195-1197 that define the standard as requiring showing of "relies on the judicial process to the detriment of the other party."

waiver-conduct → waiver-standard, waiver-mere, waiver-expense

Key finding from Davis v. Continental Airlines (1997), chunk id 598547 (opinion_id 2243859, chunk 8): "There is no single test for waiver of the right to compel arbitration, but waiver may be found where the party seeking arbitration has (1) previously taken steps inconsistent with an intent to invoke arbitration, (2) unreasonably delayed in seeking arbitration, or (3) acted in bad faith or with willful misconduct. (Christensen v. Dewor Developments, supra, 33 Cal.3d at p. 782, citing Keating v. Superior Court, supra, 31 Cal.3d at p. 605.) The moving party's mere participation in litigation is not enough; the party who seeks to establish waiver must show that some prejudice has resulted from the other party's delay in seeking arbitration. (Keating v. Superior Court, supra, at pp. 605-606.)"

This directly answers the second part of the question - mere participation in litigation is NOT enough. Now looking for more modern cases with the same standard.

waiver-participation → waiver-mere, waiver-standard

Great findings. I'm also looking for the specific standard from the Keating case, and I found that Roman v. Superior Court (2009) directly addresses the issue of whether litigation expenses are sufficient to show waiver. The key language: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses."

Let me see if anyone has found chunk id 275136 from St. Agnes yet.

waiver-history → waiver-standard, waiver-cases, waiver-conduct, waiver-mere, waiver-SC, waiver-CA

Key authorities identified so far. Please check for additional cases. The core answer is in St. Agnes Medical Center v. PacifiCare of California (2003) 31 Cal.4th 1187, opinion_id 2519923. Key chunks from it:

- chunk_id 5 (id 304893): heavy burden of proof
- chunk_id 7 (id 151628): 6-factor test
- chunk_id 22 (id 275136): "waiver does not occur by mere participation in litigation" + "courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses"

Also key: Roman v. Superior Court (2009) 172 Cal.App.4th 1462, opinion_id 2287958, which applies St. Agnes and says mere litigation expenses are insufficient for waiver.

Please search for and read: (1) Groom v. Health Net (opinion_id 2267944), (2) Engalla v. Permanente Medical Group (California Supreme Court), (3) Sobremonte v. Superior Court (opinion_id 2258122), and (4) Christensen v. Dewor Developments (33 Cal.3d 778). Report all relevant chunk IDs.

waiver-CA → waiver-standard, waiver-mere, waiver-cases, waiver-SC, waiver-history

Key findings so far. The most important chunk IDs to report are from:

1. St. Agnes Medical Center v. PacifiCare of California (2003) 31 Cal.4th 1187 - opinion_id 2519923
- chunk 22 (id 275136) - the critical language: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses."
- chunk 7 (id 151628) - the six-factor test
- chunk 6 (id 304893) - "waivers are not to be lightly inferred and the party seeking to establish a waiver bears a heavy burden of proof"

2. Lewis v. Fletcher Jones (2012) 203 Cal.App.4th 439 - opinion_id 5666491
- chunk 8 (id 752644) - detailed statement of the six St. Agnes factors

3. Roman v. Superior Court (2009) 172 Cal.App.4th 1462 - opinion_id 2287958
- chunk 25 (id 830916) - quotes the St. Agnes standard on mere participation

Still need to find: the Engalla case (California Supreme Court) that discusses the "preponderance of the evidence" standard; the Christensen v. Dewor Developments case. Searching now.

waiver-conduct → waiver-history

The key chunk id to report is 275136 (Saint Agnes chunk 22) which directly answers the question. Let me search for the specific fact pattern: employer answers complaint, pays jury fees, serves/answers discovery, meet-and-confer, takes deposition ~1 year later. Is there a specific California appellate case matching those exact facts?

waiver-1281 → waiver-standard, waiver-cases, waiver-conduct, waiver-mere, waiver-expense, waiver-SC, waiver-CA, waiver-morgan, waiver-history, waiver-participation

Key findings so far. The most important authority is St. Agnes Medical Center v. PacifiCare of California (2003) 31 Cal.4th 1187 (opinion id 2519923). Chunk 22 (id 275136) contains the key holding: "In California, whether or not litigation results in prejudice also is critical in waiver determinations." It states: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses." Cites Groom v. Health Net (2000) 82 Cal.App.4th 1189 for the point that mere expense of responding to preliminary matters is not enough.

Still need to find: the specific case in the behavioral/social sciences that matches the fact pattern in the question (employer answers complaint, pays jury fees, does discovery, takes deposition, moves to compel ~1 year later). Searching for more on that now.

waiver-SC → waiver-mere, waiver-history, waiver-CA, waiver-standard, waiver-cases
Looking for specific cases matching the fact pattern: employer answers complaint, pays jury fees, written discovery served and responded to, meet-and-confer, plaintiff deposition, then moves to compel arbitration about a year after filing. Also need to find Engalla v. Permanente Medical Group (California Supreme Court) relevant chunks, and Christensen v. Dewor Developments (33 Cal.3d 778). Can anyone search for these?

waiver-morgan → waiver-mere, waiver-standard, waiver-cases, waiver-conduct, waiver-expense, waiver-1281, waiver-SC, waiver-CA, waiver-history, waiver-participation
Key findings so far on the arbitration waiver question. The St. Agnes Medical Center v. PacifiCare case (opinion_id 2519923) is the leading California Supreme Court authority. Key chunk id 275136 contains the key language: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses." I'm now looking for the specific case that matches the fact pattern in the question (employer answered complaint, paid jury fees, served and responded to written discovery, participated in meet and confer, took plaintiff's deposition, then moved to compel arbitration a year later). Can anyone find that specific case?

waiver-cases → waiver-mere, waiver-standard, waiver-conduct, waiver-expense, waiver-participation
Looking for cases most on-point for the question's specific fact pattern: employer answered complaint, paid jury fees, served and responded to written discovery, participated in meet-and-confer, took plaintiff's deposition, then moved to compel arbitration about a year later. Need (a) the specific legal standard formulation and (b) whether mere participation + expense is enough. Key cases to read in full: Groom v. Health Net (82 Cal.App.4th 1189), Guess v. Superior Court, and any 2015-2020 cases applying St. Agnes to employment arbitration waiver. Report the most helpful chunk IDs as you find them.

waiver-standard → waiver-standard, waiver-1281, waiver-SC, waiver-CA, waiver-morgan, waiver-history, waiver-participation
Ready to compile chunk IDs for reporting. Key chunks identified so far - let me confirm additional reads.

waiver-expense → waiver-cases, waiver-participation
I need to find a specific California case with these facts: employer (defendant) (1) answered a complaint, (2) paid jury fees, (3) served and responded to written discovery, (4) participated in meet-and-confer, (5) took the plaintiff's deposition, then about a year after suit filed moved to compel arbitration. The case discusses whether this constitutes waiver under St. Agnes. The question is from before Morgan v. Sundance (2022). Can you find this specific case? Try searching for "answered complaint paid jury fees written discovery deposition arbitration" in California Court of Appeal.

root → waiver-standard, waiver-cases, waiver-conduct, waiver-mere, waiver-expense, waiver-1281, waiver-SC, waiver-CA, waiver-morgan, waiver-history, waiver-participation
Key findings so far. The question asks about the California standard for waiver of right to compel arbitration, specifically whether mere participation in litigation + litigation expense is enough.

The most critical authority is St. Agnes Medical Center v. PacifiCare (2003) 31 Cal.4th 1187. Key chunk IDs:

- 952689 (chunk 6) - "State law...reflects a strong policy favoring arbitration...waivers are not to be lightly inferred and the party seeking to establish a waiver bears a heavy burden of proof"
- 151628 (chunk 7) - Sobremonte factors listed
- 506756 (chunk 8) - Sobremonte factors, question of fact
- 715810 (chunk 21) - Prejudice critical in waiver determinations
- 275136 (chunk 22) - KEY: "Because merely participating in litigation, by itself, does not result in a waiver, courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses"
- 641554 (chunk 23) - "Prejudice typically is found only where the petitioning party's conduct has substantially undermined this important public policy..."

Please help find: (1) any specific case that matches the fact pattern of employer who answered complaint, paid jury fees, served written discovery, met and conferred, took deposition, then moved to compel arbitration about a year

later; and (2) any additional sections of St. Agnes or Adolph that may be useful.

waiver-morgan → **waiver-mere**, **waiver-standard**, **waiver-cases**, **waiver-CA**, **waiver-participation**

Almost ready to report. Key chunk IDs to report (most relevant first): 275136 (St. Agnes key language), 947839 (Groom v. Health Net), 833094 (Keating v. Superior Court), 670933 (Groom second chunk), 641554 (St. Agnes analysis of prejudice examples), 970710 (Sprunk v. Prisma - quotes Sobremonte 6 factors), 147348 (Cox v. Bonni application). Are there any other key chunk IDs from your searches I should add? Also, are there any cases that match the exact fact pattern in the question (employer answered, paid jury fees, served/written discovery, meet and confer, deposition, moved to compel ~1 year later)?

waiver-cases → **waiver-standard**, **waiver-CA**, **root**

Ready to report the most helpful chunk IDs. Key chunk IDs identified:

- 275136 (St. Agnes chunk 22) - "Because merely participating in litigation..."
- 641554 (St. Agnes chunk 23) - "Prejudice typically is found only where..."
- 304893 (St. Agnes chunk 6) - heavy burden
- 151628 (St. Agnes chunk 7) - 6-factor test
- 901004 (Sobremonte chunk 12) - cites Keating/Davis standard
- 872757 (Sobremonte chunk 13) - 6-factor analysis
- 30840 (Bower chunk 14) - Hooper citation
- 554539 (Gloster chunk 12) - quotes the test
- 147348 (Cox v. Bonni chunk 21) - applies to mere participation
- 247751 (Cox v. Bonni chunk 19) - St. Agnes factors discussion
- 965307 (St. Agnes chunk 0) - intro

Confirm these are the most important IDs to report and are correct.

waiver-participation → **waiver-morgan**, coordinator

Ready to report. I've identified the key chunks most relevant to the question. The answer is: Under California law (St. Agnes Medical Center v. PacifiCare, 2003, 31 Cal.4th 1187), the standard is the multi-factor test derived from Sobremonte/Peterson: the court considers (1) conduct inconsistent with right to arbitrate, (2) substantial invocation of litigation machinery, (3) timing of arbitration motion, (4) counterclaim without motion to stay, (5) intervening steps like judicial discovery, (6) whether delay affected, misled, or prejudiced opposing party. Critically, mere participation in litigation plus litigation expense is NOT enough — "courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses."

waiver-standard → **waiver-mere**, **waiver-cases**

I'm about to report helpful IDs. Do you have any additional chunk IDs you found that I haven't seen? Key ones I want to include: 275136, 952689, 151628, 506756, 641554, 947839, 830916, 554539, 821587, 539378, 437226, 222524, 443541, 372794, 617984, 149945. Please add any more.

waiver-CA → **waiver-morgan**, **waiver-standard**, **waiver-SC**

Compiling final list of chunk IDs to report now. The question is about California's pre-Morgan v. Sundance standard for waiver of the right to compel arbitration. The answer is: (1) The standard is the St. Agnes six-factor multifactor test; a party resisting arbitration bears a heavy burden; (2) No, mere participation in litigation plus litigation expense is not enough - St. Agnes flatly holds that "courts will not find prejudice where the party opposing arbitration shows only that it incurred court costs and legal expenses."

Key ids I'm reporting with highest priority:

- 275136 (St. Agnes chunk 22 - the "merely participating" language)
- 151628 (St. Agnes chunk 7 - 6-factor test)
- 952689 (St. Agnes chunk 6 - heavy burden)
- 715810 (St. Agnes chunk 21 - prejudice determinative)
- 678878 (Groom v. Health Net - mere expense not enough)
- 430435 (Zamora - quoting St. Agnes)
- 830916 (Roman v. Superior Court - quoting St. Agnes)
- 752644 (Lewis v. Fletcher Jones - 6 factors)

Are there more ids I should include?

waiver-cases → **waiver-standard**

Adding these important chunk IDs to the report list:

- 833094 (Keating v. Superior Court chunk 23) - the foundational statement "Waiver does not occur by mere participation in litigation"
- 678878 (Groom v. Health Net chunk 13) - "Groom cites no case that holds that the mere expense of responding to preliminary court motions, by itself, constitutes the prejudice that bars a belated petition to compel arbitration"
- 377873 (Groom v. Health Net chunk 9) - burden of proof discussion
- 395382 (St. Agnes chunk 24) - application to facts

Please include these in your report.

`waiver-expense` → `root`, `waiver-standard`, `waiver-mere`

Ready to report. The most relevant chunk IDs for the question about the California standard for waiver of right to compel arbitration (pre-Morgan v. Sundance) and whether mere participation + expense is enough are:

- 275136 (St. Agnes chunk 22) - key language
- 833094 (Keating chunk 23) - key language
- 641554 (St. Agnes chunk 23) - prejudice standards
- 678878 (Groom v. Health Net chunk 13) - mere expense not enough
- 736262 (Dunham chunk 8) - legal standard
- 194514 (CCP § 1281.2)
- 151628 (St. Agnes chunk 7) - 6 factors
- 598547 (Davis v. Continental chunk 8) - factors
- 832174 (Nunez v. Nevell chunk 9) - factors restated

B. Performance by Corpus

The validation mix of Figure 1 has 506 queries over 19 corpora. Tables 2, 3 and 4 break its NDCG, cost and estimated inference time down by corpus. They cover the systems of Figure 1 at the reasoning effort shown there, plus GPT-6 Luna at medium effort. DeepSeek V4.1 Flash's high effort is also what DeepInfra sends for medium. The last row of each table, All, holds the numbers of Figure 1.

Queries. 207 of the 506 queries come from public benchmarks: 20 each from HotpotQA (Yang et al., 2018), FiQA-2018 (Maia et al., 2018) and SciFact (Wadden et al., 2020) as packaged in BEIR (Thakur et al., 2021), from the Enron email questions of ART·E (Corbitt, 2025), from Housing Statute QA and Bar Exam QA (Zheng et al., 2025b), and from WixQA (Cohen et al., 2025), BrowseComp-Plus (Chen et al., 2025) and EnterpriseRAG-Bench (Sun et al., 2026); 11 from FRAMES (Krishna et al., 2025), answered on HotpotQA's Wikipedia corpus; and 16 from OBLIQ-Bench (Tchuindjo et al., 2026), whose four corpora (congressional hearings, tweets, chat logs and mathematical proofs) share one row. The rest of the questions were generated with our proposer pipeline. Every model was given an NDCG objective prompt.

Cost. Cost is token usage times list price. GPT-6 Astra's total was priced request by request, with long-context rates, and is divided among its queries in proportion to their token usage. SID-2 Swarm is priced at $0.50 input, $0.05 cached input and $2.00 output per million tokens, on the model calls of all its agents. Its cost assumes a perfect prefix cache, as its latency model does: each call pays the input rate only for its new tokens, such as tool results, messages or a forked agent's own instructions, and the cached rate for the rest of its context. Harness-1 ran on our own GPU. It is priced at DeepInfra's list prices: $0.03 input and $0.14 output per million tokens for gpt-oss-20b, and $0.05 per million tokens for the Qwen3-Reranker-8B calls its search and read tools make. These average 17 calls per query, about 25k tokens each (estimated), so the reranker accounts for $21.6 of its $32.6 per 1,000 queries.

Inference time. Figure 1 and Table 4 place each system by its estimated model inference time per query, not its measured time. The runs come from different days, providers, deployments and concurrencies, so their wall-clock times do not compare; an estimate from each run's token usage does. Retrieval backend time is excluded for every system, and vector search and BM25 alone have no model inference. For the API and open-weight models, the time per query is $n_{\text{calls}} \cdot t_{\text{first}} + n_{\text{out}}/v_{\text{out}}$. Here n_{calls} is the number of model calls and n_{out} the number of output tokens, reasoning included. t_{first} and v_{out} are the median time to first token and output speed that Artificial Analysis reports for one provider (Table 1). SID-2's time comes from the latency

model of §3.3, on one B200 at 8 concurrent queries. The vector search + reranker baseline in Figure 1 uses DeepInfra's server time for scoring 100 candidates with Qwen3-Reranker-8B. Per corpus, the same estimates are applied to each query's own model calls and tokens.

System	Artificial Analysis provider	First token (s)	Output (tok/s)
GPT-6 Astra	OpenAI	1.00[a]	52.6
GPT-5.6 Luna	OpenAI	1.00[a]	114.8
GPT-6 Luna, every effort	OpenAI, high effort[b]	1.00[a]	127.0
Opus 5	Anthropic	1.00[a]	49.5
Opus 5.5	Anthropic, medium effort	1.00[a]	75.0
Kimi K3	Modal	0.89	92.0
GLM-5.3	Z.ai	2.95	67.0
Qwen3.8 Max	Alibaba	2.82	40.4
DeepSeek V4.1 Flash, every effort	DeepInfra, max effort[b]	0.75	73.0
Gemma 4 31B	DeepInfra	2.95	17.0
Gemma 4 26B-A4B	DeepInfra (FP8)	0.82	31.5
Harness-1 (gpt-oss-20b)	gpt-oss-20B, median across providers	0.78	132.0

Table 1 | Speeds behind the time estimates of Figure 1. [a]OpenAI and Anthropic hide reasoning, so Artificial Analysis's time to first token measures the first answer token; we assume 1 s instead. [b]No per-effort figure is published, so every effort uses this one.

Results. SID-2 Swarm has the highest NDCG on 11 of the 19 corpora (9 with twelve agents, 2 with six), GPT-6 Astra (max) on 6. The Swarm's gain over a single SID-2 agent at high effort is largest where a query needs many searches: +0.42 on BrowseComp-Plus, +0.28 on arXiv papers and +0.23 on call transcripts. It is smallest on the corpora where most systems are near the ceiling: +0.05 on HotpotQA and SciFact, −0.01 on WixQA. SID-2 at low effort is the fastest system on all 19 corpora and the cheapest on 8. GPT-6 Luna at medium effort is the cheapest on 9.

Caveats.

- **Few queries per corpus.** Most corpora have 20 queries, the largest 66. At 20 queries, the 95% bootstrap interval of a mean NDCG is about ±0.10–0.15, so many differences within a row are not significant. The All row is the reliable comparison.
- **Approximate speeds.** The assumed time to first token and the single speed per model across efforts (Table 1) make the low-effort GPT-6 Luna and DeepSeek points the least certain.
- **Estimates differ from measurements.** GLM-5.3 and Qwen3.8 Max ran on our own deployment, where they were about twice as fast as the provider speeds assumed. The other models measured at the provider the estimate assumes came within 0.83–1.22× of their estimates. The GPT-6 Luna and DeepSeek effort sweeps ran 64 queries at a time, and Harness-1 on our own H100, so their measured times do not compare.
- **Refusals score zero.** Anthropic's safety classifier refused 33 queries to Opus 5.5 on every turn, 30 of them from SciFact, so Opus 5.5's SciFact NDCG (0.270) measures refusals rather than retrieval. With Opus 5's answers on them, Opus 5.5 would reach NDCG 0.623 instead of 0.577. OpenAI refused 2 PubMed queries to GPT-6 Astra, and GPT-6 Luna (max) exceeded its context window on 1.
- **Some models do not finish.** Almost every other unanswered query (Table 2, last row) used up the 10-turn limit while still searching; a few ended with a report that listed

no documents. This is frequent for Inkling (118 queries), Nemotron 3 Ultra (91), Beam 1.0 (90), Gemma 4 26B-A4B (69) and GLM-5.3 (55). It is concentrated on HotpotQA and BrowseComp-Plus, where it accounts for much of their gap: Inkling, Nemotron 3 Ultra and Beam 1.0 left 17, 16 and 10 of the 20 BrowseComp-Plus queries unanswered. GPT-6 Luna left 11 of the 20 PubMed queries unanswered at maximum effort and 14 at medium. SID-2 Swarm reached its episode length limit before any agent reported on 4, 1 and 0 queries with 3, 6 and 12 agents. Each tool result ends with the number of turns left out of 10, and the last turn adds "This is your last turn: you must call `report_helpful_ids` now. Anything you have found but not reported is lost when this turn ends." Every system saw the same counter and instruction. Two behaviours account for most unanswered queries:

  - *One search per turn.* Nemotron 3 Ultra and Inkling make a single tool call in 99% and 97% of their turns, and Beam 1.0 in 53%, against 13% for DeepSeek V4.1 Flash and 15% for GLM-5.3. With the same 10 turns, they reach the limit with fewer searches done and are still searching on the last turn.
  - *The last-turn instruction is ignored.* On the last turn of their unanswered queries, Nemotron 3 Ultra, Inkling and Beam 1.0 almost always searched or read again instead of reporting. Their reasoning on that turn mentions reporting in 22 of 91, 91 of 117 and 45 of 90 of these queries, typically noting that a report is due and then deciding on one more search. Beam 1.0 ended 16 of them with a text answer and no tool call. GLM-5.3 (max) ended 53 of its 54 that way: it writes its answer as text on the last turn instead of calling `report_helpful_ids`.

  The harness does not report on a model's behalf, so these queries score zero. The counts here are queries without a `report_helpful_ids` call; the totals above also include the few with an empty report.
- **Harness-1 runs its own harness.** It ends with a ranked list rather than a report, so it has no unanswered count. Its time counts only the gpt-oss-20b generation: its search and read tools also rerank with Qwen3-Reranker-8B, about 17 calls and 19 s per query on our H100, which is in its cost but not its time.
- **Gemma drops its reasoning.** The Gemma models' chat template drops reasoning between turns, which may hold them back.

		SID-2			SID-2 Swarm			Closed models				Open-weight models										Retrieval	
Corpus	Questions	Low	Medium	High	3 agents	6 agents	12 agents	GPT-6 Astra (max)	Opus 5.5 (medium)	GPT-6 Luna (max)	GPT-6 Luna (medium)	DeepSeek V4.1 Flash (high)	Kimi K3 (max)	GLM-5.3 (max)	Qwen3.8 Max (xhigh)	Beam 1.0 (max)	Harness-1 (high)	Gemma 4 31B	Inkling (max)	Nemotron 3 Ultra (high)	Gemma 4 26B-A4B	Vector search	BM25
Corpora of public benchmarks																							
HotpotQA	66	0.843	0.868	0.877	0.932	**0.951**	0.931	0.933	0.907	0.910	0.867	0.899	0.925	0.896	0.869	0.685	0.785	0.777	0.629	0.568	0.552	0.376	0.385
FiQA	41	0.541	0.571	0.583	0.657	0.652	0.677	**0.761**	0.577	0.584	0.602	0.671	0.648	0.545	0.543	0.551	0.435	0.471	0.387	0.392	0.361	0.307	0.277
SciFact	43	0.893	0.894	0.884	0.911	0.936	0.930	0.941	0.270	**0.950**	0.916	0.920	0.935	0.841	0.925	0.865	0.805	0.843	0.769	0.774	0.735	0.700	0.707
Enron emails	49	0.585	0.600	0.600	0.654	0.652	**0.661**	0.633	0.594	0.349	0.310	0.543	0.581	0.597	0.365	0.578	0.510	0.349	0.441	0.253	0.247	0.439	0.468
HousingQA	21	0.564	0.566	0.545	0.601	0.601	0.618	**0.645**	0.590	0.550	0.560	0.507	0.570	0.536	0.537	0.408	0.386	0.471	0.276	0.328	0.344	0.068	0.180
BarExamQA	20	0.154	0.164	0.177	0.229	**0.286**	0.280	0.160	0.164	0.000	0.025	0.215	0.156	0.172	0.170	0.166	0.135	0.165	0.032	0.000	0.151	0.132	0.000
WixQA	20	0.876	0.859	0.901	0.897	0.922	0.889	0.901	0.886	0.814	0.862	0.886	0.902	**0.933**	0.924	0.835	0.749	0.762	0.820	0.844	0.816	0.785	0.401
BrowseComp-Plus	20	0.483	0.474	0.484	0.632	0.714	0.906	**0.971**	0.695	0.667	0.527	0.865	0.851	0.657	0.655	0.134	0.453	0.170	0.082	0.090	0.160	0.033	0.055
EnterpriseRAG-Bench	20	0.770	0.789	0.854	0.930	0.919	**0.938**	0.912	0.856	0.845	0.828	0.879	0.840	0.822	0.890	0.863	0.643	0.589	0.819	0.572	0.532	0.546	0.621
OBLIQ-Bench	16	0.045	0.054	0.043	0.052	0.137	0.228	**0.239**	0.132	0.128	0.103	0.133	0.154	0.075	0.078	0.061	0.015	0.033	0.035	0.029	0.029	0.016	0.009
Our corpora																							
arXiv papers	27	0.425	0.465	0.463	0.652	0.702	**0.745**	0.655	0.501	0.343	0.332	0.538	0.375	0.326	0.400	0.460	0.284	0.242	0.220	0.248	0.188	0.138	0.123
arXiv passages	13	0.296	0.338	0.314	0.472	0.495	**0.539**	0.465	0.394	0.326	0.225	0.453	0.329	0.294	0.322	0.310	0.042	0.193	0.203	0.116	0.189	0.164	0.124
California law	20	0.247	0.274	0.337	0.488	0.506	0.522	**0.549**	0.401	0.171	0.035	0.419	0.472	0.316	0.266	0.157	0.156	0.224	0.070	0.056	0.060	0.060	0.036
Call transcripts	20	0.597	0.662	0.694	0.877	0.888	**0.930**	0.920	0.639	0.644	0.522	0.744	0.729	0.624	0.551	0.428	0.498	0.358	0.352	0.288	0.317	0.301	0.217
LinkedIn profiles	20	0.464	0.475	0.494	0.606	0.670	**0.681**	0.450	0.630	0.464	0.411	0.534	0.531	0.465	0.450	0.442	0.255	0.269	0.323	0.283	0.270	0.156	0.063
PubMed	20	0.167	0.194	0.227	0.297	0.310	**0.325**	0.252	0.250	0.097	0.035	0.180	0.174	0.181	0.138	0.207	0.137	0.092	0.107	0.067	0.063	0.056	0.030
SEC filings	20	0.521	0.661	0.606	0.728	0.784	0.786	**0.840**	0.724	0.728	0.634	0.719	0.712	0.651	0.636	0.478	0.196	0.331	0.420	0.381	0.130	0.055	0.078
Wikidata companies	30	0.422	0.479	0.498	0.561	0.607	**0.637**	0.569	0.528	0.498	0.405	0.521	0.470	0.365	0.404	0.390	0.287	0.275	0.224	0.254	0.182	0.136	0.036
YC companies	20	0.529	0.611	0.616	0.723	0.732	**0.746**	0.728	0.684	0.670	0.569	0.667	0.666	0.589	0.599	0.513	0.296	0.180	0.358	0.314	0.251	0.100	0.083
All	506	0.556	0.585	0.594	0.676	0.701	**0.720**	0.704	0.577	0.565	0.520	0.640	0.632	0.575	0.558	0.505	0.444	0.424	0.398	0.356	0.337	0.287	0.260
Unanswered[a]		0.3	1	0.3	4	1	0	11	36	18	20	10	4	55	2	90	–	14	118	91	69	–	–

Table 2 | NDCG per corpus of the 506-query validation mix (higher is better). Best value in each row in bold, second best underlined. SID-2 is the mean of three runs, every other system a single run. [a]Queries (of 506) that ended without a report; they score 0. For SID-2, the mean over its three runs. Harness-1 runs its own harness, which has no report tool.

		SID-2			SID-2 Swarm			Closed models				Open-weight models									
Corpus	Questions	Low	Medium	High	3 agents	6 agents	12 agents	GPT-6 Astra (max)	Opus 5.5 (medium)	GPT-6 Luna (max)	GPT-6 Luna (medium)	DeepSeek V4.1 Flash (high)	Kimi K3 (max)	GLM-5.3 (max)	Qwen3.8 Max (xhigh)	Beam 1.0 (max)	Harness-1 (high)	Gemma 4 31B	Inkling (max)	Nemotron 3 Ultra (high)	Gemma 4 26B-A4B
Corpora of public benchmarks																					
HotpotQA	66	2.92	3.71	4.92	82.3	157	313	211	35.4	2.51	**1.38**	7.24	112	86.3	45.8	6.95	26.1	5.48	17.7	9.51	4.08
FiQA	41	**2.04**	3.77	6.98	140	280	567	1,847	84.5	6.93	2.87	15.6	207	76.8	72.2	18.4	35.5	5.28	32.6	12.9	3.89
SciFact	43	3.52	5.71	7.31	118	228	439	759	14.6	6.15	**2.64**	25.7	224	133	74.9	17.4	34.6	7.21	40.2	17.3	4.67
Enron emails	49	4.10	6.57	8.03	103	217	426	754	85.3	5.26	**1.98**	24.9	111	84.0	54.1	26.0	18.6	9.78	43.0	15.9	4.76
HousingQA	21	3.93	8.23	11.1	157	305	602	1,178	84.5	8.09	**3.66**	22.9	205	159	96.5	17.5	38.4	10.6	72.7	21.7	4.96
BarExamQA	20	**2.75**	5.82	10.5	141	263	527	1,246	43.1	7.62	3.00	20.4	308	145	66.9	19.1	34.6	7.59	42.3	15.5	6.50
WixQA	20	**1.54**	2.62	4.89	84.9	178	353	1,133	44.5	4.94	2.05	7.07	97.1	59.0	42.1	14.1	36.5	2.54	22.8	8.94	2.84
BrowseComp-Plus	20	14.7	15.9	18.3	227	438	764	1,493	116	16.9	10.3	33.7	800	256	159	20.8	36.8	19.0	68.5	31.4	**8.36**
EnterpriseRAG-Bench	20	3.55	5.59	8.29	134	264	517	1,512	94.0	12.2	6.14	27.8	157	85.4	46.5	20.6	30.6	5.55	35.8	12.4	**3.39**
OBLIQ-Bench	16	**5.09**	8.07	11.6	196	347	693	2,730	184	9.28	5.10	34.3	367	271	134	29.2	35.4	12.1	56.7	20.0	5.27
Our corpora																					
arXiv papers	27	5.57	7.75	10.4	141	286	552	5,126	440	14.2	**5.29**	47.8	463	286	177	77.6	35.1	9.21	88.5	21.6	6.37
arXiv passages	13	4.54	8.09	11.9	170	324	646	3,424	166	9.95	**3.27**	31.6	362	210	130	32.8	38.3	6.99	55.2	16.9	5.75
California law	20	6.23	10.2	15.2	202	380	747	2,501	226	10.9	**4.23**	42.8	600	337	172	29.5	38.9	14.7	80.2	29.6	8.31
Call transcripts	20	3.29	6.05	10.0	145	295	609	1,521	117	6.78	**2.66**	16.0	159	90.2	78.1	16.7	37.4	7.69	37.3	18.7	4.69
LinkedIn profiles	20	**5.45**	8.62	10.5	152	297	574	7,265	378	44.2	15.0	48.5	529	356	172	81.9	32.8	10.6	73.9	17.4	5.64
PubMed	20	4.39	7.56	12.1	143	281	549	4,099	122	3.22	**1.21**	15.4	291	138	112	29.5	37.5	5.93	60.8	18.3	4.92
SEC filings	20	**6.86**	8.70	12.6	188	350	680	2,723	181	16.6	6.97	24.6	529	269	145	36.2	37.0	10.8	67.5	31.6	7.04
Wikidata companies	30	**3.03**	5.55	7.73	110	216	427	3,845	138	15.0	3.71	29.4	320	199	92.5	29.1	32.1	7.53	24.9	10.9	3.24
YC companies	20	**4.91**	6.93	9.00	142	269	540	4,571	310	41.0	17.0	40.8	422	342	164	96.0	33.5	8.82	68.6	14.6	6.48
All	506	**4.28**	6.54	9.15	135	263	515	2,141	131	10.9	4.44	24.9	288	167	95.7	29.2	32.6	8.33	47.0	16.9	5.06

Table 3 | Cost per 1,000 queries in USD, per corpus (lower is better): token usage × list price. Best value in each row in bold, second best underlined. SID-2 is the mean of three runs, every other system a single run. SID-2 Swarm at $0.50 input, $0.05 cached input and $2.00 output per million tokens, with a perfect prefix cache. Harness-1 ran on our own GPU and is priced at DeepInfra's list prices for gpt-oss-20b and for the Qwen3-Reranker-8B calls of its tools (tokens per call estimated). Retrieval alone is left out.

		SID-2			SID-2 Swarm			Closed models				Open-weight models									
Corpus	Questions	Low	Medium	High	3 agents	6 agents	12 agents	GPT-6 Astra (max)	Opus 5.5 (medium)	GPT-6 Luna (max)	GPT-6 Luna (medium)	DeepSeek V4.1 Flash (high)	Kimi K3 (max)	GLM-5.3 (max)	Qwen3.8 Max (xhigh)	Beam 1.0 (max)	Harness-1 (high)	Gemma 4 31B	Inkling (max)	Nemotron 3 Ultra (high)	Gemma 4 26B-A4B
Corpora of public benchmarks																					
HotpotQA	66	**1.46**	2.09	3.11	11.7	13.4	16.6	31.3	14.5	25.1	14.2	75.4	44.2	175	107	11.4	74.0	278	21.1	27.7	244
FiQA	41	**1.23**	2.21	4.22	14.5	16.4	22.1	120	24.9	43.6	19.0	112	70.1	143	127	18.6	87.8	277	25.7	31.1	214
SciFact	43	**1.60**	2.67	3.97	14.2	16.6	21.4	49.5	10.8	35.6	15.3	115	54.3	265	147	16.8	86.6	342	26.5	32.3	272
Enron emails	49	**1.54**	2.62	3.67	9.53	12.6	17.1	43.8	13.1	28.1	10.0	78.5	31.0	123	68.3	14.1	50.7	175	17.0	19.9	142
HousingQA	21	**1.63**	3.19	5.02	14.6	18.0	24.1	83.6	21.3	45.8	19.7	91.6	68.6	287	132	15.7	92.7	335	27.9	32.6	188
BarExamQA	20	**1.46**	3.04	5.86	15.4	17.4	22.7	124	13.3	51.3	20.4	99.0	78.9	282	129	22.0	91.1	433	38.0	36.3	379
WixQA	20	**0.93**	1.52	2.80	9.31	11.0	14.9	74.7	13.7	33.2	13.2	59.2	36.2	98.1	63.4	14.6	87.2	125	19.6	22.6	140
BrowseComp-Plus	20	**4.71**	5.82	7.80	21.2	25.5	30.0	74.9	37.5	55.3	32.8	136	124	475	331	19.3	101	767	39.5	45.6	509
EnterpriseRAG-Bench	20	**1.58**	2.52	3.80	12.2	14.8	20.4	56.2	16.7	30.4	15.0	78.5	44.4	139	69.3	16.0	78.9	186	19.5	21.0	146
OBLIQ-Bench	16	**2.28**	3.91	6.21	25.0	25.1	34.0	152	36.4	49.7	26.5	186	149	556	265	32.2	99.7	493	52.8	40.5	299
Our corpora																					
arXiv papers	27	**2.48**	3.62	5.29	17.7	22.1	29.3	255	41.3	68.7	25.6	232	126	469	329	44.8	90.5	384	62.8	38.4	271
arXiv passages	13	**1.95**	3.44	5.58	15.2	18.9	26.3	318	32.7	57.0	19.1	123	115	378	220	22.7	103	304	26.6	30.9	235
California law	20	**2.46**	4.24	6.92	16.0	18.7	24.7	154	47.8	55.3	22.8	110	169	631	239	17.7	97.9	390	32.0	38.1	244
Call transcripts	20	**1.53**	2.86	5.18	14.8	18.3	24.8	109	26.5	47.5	19.2	81.8	60.0	170	111	14.3	92.8	297	25.3	31.3	185
LinkedIn profiles	20	**2.66**	4.52	6.40	18.2	22.8	27.8	263	47.5	159	36.7	211	172	533	299	63.6	78.8	365	45.9	33.2	219
PubMed	20	**1.98**	3.44	5.80	13.9	18.6	25.2	231	27.1	33.8	19.6	90.4	93.6	254	178	24.2	90.5	256	28.5	28.2	209
SEC filings	20	**2.82**	4.15	6.83	20.4	22.5	28.1	174	34.7	76.4	28.1	149	111	473	245	20.9	95.6	373	26.8	40.8	291
Wikidata companies	30	**2.12**	3.69	5.63	17.6	20.5	26.3	381	33.1	101	34.6	199	134	408	212	38.7	82.5	347	29.9	32.6	156
YC companies	20	**2.57**	3.82	5.68	19.4	21.2	29.4	276	38.4	137	33.7	218	135	453	311	52.3	83.5	412	48.0	41.4	294
All	506	**1.92**	3.11	4.86	15.0	17.7	23.1	135	25.1	53.8	20.8	122	84.7	296	171	23.2	83.9	325	30.1	31.6	236

Table 4 | Estimated model inference time per query in seconds, per corpus (lower is better). Retrieval backend time is excluded. Best value in each row in bold, second best underlined. SID-2 is the mean of three runs, every other system a single run. Vector search and BM25 have no model inference and are left out.